FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Alternativa a Entidades Certificadoras de Chaves Públicas

Flávio Francisco Pinto Moreira da Silva

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Professor José Manuel de Magalhães Cruz

14 de Julho de 2017

Resumo

O contínuo aumento de serviços sustentados na Internet justifica ainda mais a necessidade de um controlo acrescido sobre todos os aspetos relativos à segurança das comunicações entre as entidades intervenientes. De entre os diversos aspetos, este trabalho centra-se na distribuição de chaves criptográficas públicas. Nesse sentido, tendo em conta a existência de vários problemas associados às Entidades Certificadoras, considera-se questionável a utilização corrente deste tipo de entidades para validar a distribuição de chaves criptográficas públicas, pelo que se entende a necessidade de uma mudança de paradigma dos métodos de validação e distribuição.

Os principais objetivos deste trabalho consistem em realizar uma caracterização detalhada do estado da arte da área juntamente com a descrição de uma proposta concreta, fundamentada e exequível, que constitua uma alternativa ou uma diminuição do papel das Entidades Certificadoras convencionais na validação de chaves criptográficas públicas. Por fim, considera-se propício realizar um conjunto de diretrizes para a implementação da proposta encontrada na infraestrutura da Universidade do Porto (U.Porto).

Na primeira etapa do trabalho são retratados os aspetos conceptuais relevantes ao tema do documento, seguida de uma sucinta descrição de um conjunto de otimizações/alternativas ao sistema atual de distribuição de chaves públicas, culminando com uma exposição do problema a tratar.

Na segunda etapa é realizada a descrição da implementação de uma estrutura de chaves públicas interna através da utilização da biblioteca *openSSL* com o intuito de provar a possibilidade do funcionamento deste tipo de sistemas sem a necessidade de recorrer a CAs externas, seguida de uma análise ao sistema de certificação digital utilizado na U.Porto.

Posteriormente são apresentadas três análises relativas a otimizações/alternativas que se consideram as mais aptas para a concretização dos objetivos do trabalho, com o intuito de providenciar um termo de comparação para a eleição de uma delas para a proposta final.

Na etapa final é detalhada uma proposta concreta da otimização/alternativa escolhida, através de uma prova de conceito relativa à sua utilização, incluindo a apresentação de um conjunto de sugestões para a sua implementação na U.Porto e da exposição das suas diversas características mediante um grupo de aspetos considerados relevantes para avaliar a validade de uma solução.

No âmbito da dissertação foram encontradas duas soluções para o problema descrito: uma parcial e outra global.

Relativamente à solução parcial, concluiu-se que instituições denotadas de uma posição social têm a possibilidade da gestão das chaves públicas das suas entidades internas e que a sua disponibilização para o exterior pode ser concretizada através de validações cruzadas entre instituições ou por Entidades Certificadoras de nível mundial. A implementação realizada permitiu provar a viabilidade da instalação de uma infraestrutura de chaves públicas numa rede local.

A solução global refere-se à utilização a nível geral do protocolo DANE, *DNS-based Authentication of Named Entities*, cobrindo os certificados digitais mas também a possibilidade da sua utilização para chaves públicas singulares. A sua prova de conceito permitiu provar a viabilidade da ativação das ferramentas necessárias ao protocolo DANE numa rede local.

Abstract

The continued growth of services supported by the Internet justifies the need for an increased control over the aspects of communications security between the intervening entities. Among the various aspects, this work focuses on the distribution of public cryptographic keys. In this sense, regarding the existence of several problems associated with Certificate Authorities, it is considered questionable the current use of this type of entities to validate the distribution of public cryptographic keys, which causes the need to a paradigm shift of validation and distribution methods.

The main objectives of this work are to carry out a detailed characterization of the state of the art of the area together with the description of a concrete, reasoned and feasible proposal that constitutes an alternative or a reduction of the role of conventional Certificate Authorities in the validation of public cryptographic keys. Finally, it is considered propitious to carry out a set of guidelines for the implementation of the proposal found in the infrastructure of Universidade do Porto (U.Porto).

In the first stage of the work are described the conceptual aspects relevant to the theme of the document, followed by a succinct description of a set of optimizations/alternatives to the current public key distribution system, culminating with an exposition of the problem to be addressed.

The second stage describes the implementation of an internal public key structure through the use of the *openSSL* library in order to prove the possibility of operating this type of system without the need of external CAs, followed by an analysis of the digital certification system used in U.Porto.

Subsequently, three analyzes are presented concerning optimizations/alternatives that are considered the most suitable for the accomplishment of the objectives of the work, in order to provide a comparison term for the election of one of them for the final proposal.

In the final step, a concrete proposal of the optimization/alternative chosen is presented, through a proof of concept regarding its use, including the presentation of a set of suggestions for its implementation at U.Porto and the exposition of its various characteristics through a group of aspects considered relevant to evaluate the validity of a solution.

In the scope of the dissertation two solutions were found for the described problem: one partial and one global.

Regarding the partial solution, it was concluded that institutions denoted by a social position have the possibility of managing the public keys of their internal entities and that their availability abroad can be accomplished through cross validations between institutions or by world-class Certificate Authorities. The implementation made it possible to prove the feasibility of installing a public key infrastructure on a local network.

The overall solution refers to the general use of the DANE protocol, *DNS-based Authentication of Named Entities*, covering the digital certificates but also the possibility of its use for singular public keys. The proof of concept allowed to prove the viability of the activation of the necessary tools to the protocol DANE in a local network.

Agradecimentos

Esta dissertação significa o culminar de uma fase crucial da minha vida e não seria possível sem a ajuda dos meus familiares e amigos. Tendo isso em mente, gostaria de agradecer a todos eles pelo acompanhamento, pelo suporte, pela solidariedade e palavras de incentivo. Por fim, gostaria de deixar um especial agradecimento ao meu orientador, o Professor José Manuel Cruz pela imensa compreensão e todo o apoio disponibilizado.

Flávio Francisco Pinto Moreira da Silva

“Believe you can and you’re halfway there.“

Theodore Roosevelt

Conteúdo

Lista de Figuras

Lista de Tabelas

Abreviaturas e Símbolos

3DES	*Triple Data Encryption Standard*
AES	*Advanced Encryption Standard*
CA	*Certification Authority*
CSR	*Certificate Signing Request*
CT	*Certificate Transparency*
DES	*Data Encryption Standard*
DNS	*Domain Name System*
DNSSEC	*Domain Name System Security Extensions*
ECDSA	*Elliptic Curve Digital Signature Algorithm*
EJBCA	*Enterprise Java Beans Certificate Authority*
EFF	*Electronic Frontier Foundation*
FEUP	Faculdade de Engenharia da Universidade do Porto
HPKP	*HTTP Public Key Pinning*
HTTP	*Hypertext Transfer Protocol*
IANA	*Internet Assigned Numbers Authority*
ICANN	*Internet Corporation for Assigned Names and Numbers*
IP	*Internet Protocol*
IPsec	*Internet Protocol Security*
LDAP	*Lightweight Directory Access Protocol*
MAC	*Message Authentication Code*
MIME	*Multipurpose Internet Mail Extensions*
MITM	*Man In the Middle*
MMD	*Maximum Merge Delay*
MTA	*Mail Transfer Agent*
OCSP	*Online Certificate Status Protocol*
PGP	*Pretty Good Privacy*
PKI	*Public Key Infrastructure*
SMTP	*Simple Mail Transfer Protocol*
SPKI	*Simple Public Key Infrastructure*
SCT	*Signed Certificate Timestamp*
SSH	*Secure Shell*
SSL	*Secure Sockets Layer*
STH	*Signed Tree Head*
TACK	*Trust Assertions for Certificate Keys*
TCS	*Trusted Certificate Service*
TLD	*Top Level Domain*
TLS	*Transport Layer Security*
TSK	*Tack Signing Key*
UA	*User Agent*
U.Porto	Universidade do Porto
URL	*Uniform Resource Locator*
VPN	*Virtual Private Network*

Capítulo 1

Introdução

1.1 Enquadramento

O contínuo aumento de transações de dados e o crescimento do número de negócios suportados em plataformas *online* suscitaram a intensificação das preocupações com a segurança informática e a procura de soluções seguras para as comunicações entre as entidades intervenientes.

Uma das principais preocupações consiste na dificuldade que uma entidade tem em identificar corretamente a entidade com quem está a trocar informações. Essa dificuldade, em termos práticos, pode ser ultrapassada através da utilização de protocolos de autenticação de criptografia assimétrica e traduz-se na confiança em corretamente mapear chaves públicas e entidades comunicadoras.

A solução mais comummente adotada para alcançar a supracitada identificação consiste em utilizar Entidades Certificadoras, em inglês *Certification Authorities* (CAs), para assegurar o válido mapeamento de uma entidade à sua chave criptográfica pública. Todas estas entidades estão inseridas num modelo denominado de Infraestrutura de Chaves Públicas que promove a legitimidade das comunicações através da cooperação de um conjunto de ferramentas e técnicas específicas.

De modo a estabelecer um canal de comunicação seguro, um utilizador necessita de conhecer a chave pública da entidade com quem deseja comunicar. Para que essa chave seja distribuída com segurança é por norma emitido um certificado digital (por uma CA) que comprova a associação entre a entidade e a chave em questão. Através deste certificado e mediante a confiança depositada nessa entidade certificadora, terceiros podem considerar como válida a associação entidade - chave pública.

A utilização de um canal de comunicação sem os mecanismos de segurança referentes à troca de chaves públicas pode fazer com que uma entidade mal-intencionada se infiltre numa comunicação fazendo-se passar pela outra parte comunicante.

A confiança nas CAs consiste na peça chave de todo modelo, o que provoca que qualquer comprometimento da sua integridade ponha em causa a validade dos certificados e consequentemente a autenticação das entidades comunicantes.

1.2 Motivação

O recurso a Entidades Certificadoras é considerado uma inevitabilidade pela generalidade dos utilizadores. Na verdade esta solução, que nos dias de hoje é imprescindível para o normal funcionamento de diversas tecnologias, conta com várias vulnerabilidades no que diz respeito a questões de segurança, de eficiência e de custos.

Tendo em conta o papel das CAs no funcionamento do modelo e todas as suas fragilidades, considera-se que este tipo de entidades possui uma responsabilidade exagerada colocando até em causa a legitimidade dos serviços e a segurança dos utilizadores. Estas razões evidenciam a necessidade de propostas que alterem o paradigma de estabelecimento de relações de confiança para a troca segura de chaves, diminuindo ou eliminando o papel das entidades certificadoras de chaves criptográficas públicas.

De outro ponto de vista, considera-se controversa a utilização de CAs externas em infraestruturas dirigidas por entidades que pela sua posição social possuem um estatuto confiável. Neste tipo de infraestruturas englobam-se estruturas governamentais, académicas e até empresariais (com fins lucrativos ou não).

1.3 Objetivos

O principal objetivo desta dissertação consiste em detalhar uma proposta concreta, fundamentada e exequível que constitua uma alternativa ou uma diminuição do papel das Entidades Certificadoras convencionais na validação de chaves criptográficas públicas. A proposta tem como intuito a possibilidade da sua adaptação a uma infraestrutura concreta, que neste caso será a Universidade do Porto. Do ponto de vista de uma estrutura interna, pretende-se também documentar a viabilidade da instalação de um sistema de certificação de chaves públicas para uma rede interna sem a necessidade de CAs externas.

1.4 Estrutura do Documento

Este documento encontra-se organizado em oito capítulos.

O Capítulo 1 tem como principal objetivo descrever a conjuntura do problema a tratar, conferindo uma contextualização do tema e uma apresentação da motivação e dos principais objetivos.

O Capítulo 2 providencia uma revisão dos temas relativos aos conceitos fundamentais de criptografia e infraestrutura de chaves públicas. Cada um destes temas é tratado com o nível de detalhe que permita prover o suporte teórico necessário para a apreensão dos assuntos a ser tratados.

O Capítulo 3 realiza a caracterização do problema em questão. Numa primeira fase são expostas as principais limitações das Entidades Certificadoras seguidas de dois exemplos de ataques reais a este tipo de entidades explicitando o método utilizado e as suas principais consequências.

O Capítulo 4 descreve a implementação realizada referente a uma estrutura de chaves públicas internas através da biblioteca *openSSL*. São apresentados todos os serviços, processos e métodos utilizados para o seu funcionamento.

O Capítulo 5 analisa o sistema de certificação digital utilizado na U.Porto por intermédio de um estudo sobre o seu estado atual de certificação, da exposição de um conjunto de sugestões e um estudo sobre a possibilidade da implementação de um sistema sem certificados digitais.

O Capítulo 6 apresenta uma análise objetiva de três otimizações/alternativas, consideradas as mais ajustadas para cumprir o principal objetivo do trabalho, com o intuito de eleger uma delas para a proposta final.

O Capítulo 7 detalha uma proposta concreta de uma otimização/alternativa através de uma prova de conceito e de um conjunto de diretivas de implementação na U.Porto.

O Capítulo 8 apresenta as conclusões do trabalho seguidas de um conjunto de propostas para o trabalho futuro.

Capítulo 2

Revisão Bibliográfica

Neste capítulo pretende-se conceder um suporte teórico sobre os principais tópicos relativos ao tema central através de uma revisão das técnicas de segurança relevantes. Primeiramente são retratados os conceitos básicos de criptografia, onde são detalhados os vários esquemas de chaves, como funcionam e para que situações a sua utilização é mais vantajosa. Seguidamente é especificada a teoria que se julga imprescindível para compreender o modelo de infraestrutura de chaves públicas explicitando as suas funções, a sua organização e as suas características. Finalmente é realizada uma breve apresentação de algumas das principais alternativas existentes às Entidades Certificadoras de chaves públicas.

2.1 Conceitos Básicos de Criptografia

Nos seguintes pontos esclarecem-se os conceitos básicos utilizados para prover segurança criptográfica. As tecnologias apresentadas facultam meios para atingir alguns dos serviços que se consideram mínimos para ser possível comunicar com segurança.

2.1.1 Criptografia de Chaves Simétricas

Na criptografia de chaves simétricas, o emissor e o recetor partilham uma chave secreta que é utilizada tanto para encriptar como desencriptar.

Uma entidade emissora encripta uma mensagem com uma chave secreta que é apenas conhecida por si e pela entidade recetora. Após o envio, a entidade recetora recebe a mensagem encriptada e utiliza a mesma chave para desencriptar a mensagem. A técnica é representada graficamente na figura 2.1.

Neste tipo de criptografia é crucial que a chave utilizada se mantenha secreta pois caso uma terceira entidade consiga ter acesso a uma chave partilhada por duas entidades em comunicação, pode utilizá-la para desencriptar as mensagens trocadas (ataque passivo) ou adulterar e/ou criar mensagens (ataque ativo).

A utilização de uma chave secreta partilhada pelas entidades provoca problemas ao nível da gestão e distribuição dessa chave. A comunicação é confiável se e só se existir previamente uma

troca segura da chave e se houver uma boa gestão de chaves por parte das entidades comunicadoras.

O facto de se utilizar a mesma chave para encriptar e desencriptar permite a utilização de algoritmos classicamente simples e de grande rapidez de execução. Exemplos de algoritmos correntemente utilizados para este tipo de criptografia são o AES (*Advanced Encryption Standard*), o *Twofish* e até o 3DES (*Triple Data Encryption Standard*).

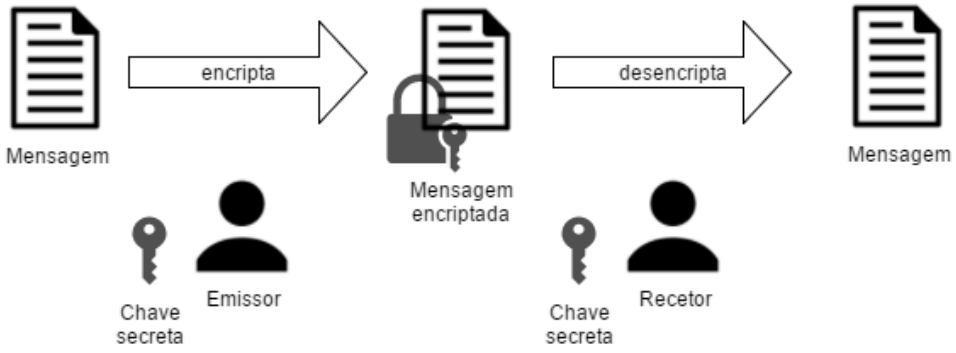


Figura 2.1: Esquema de criptografia simétrica

2.1.2 Criptografia de Chaves Assimétricas

Na criptografia de chaves assimétricas cada entidade possui um par de chaves constituído por uma chave privada, apenas conhecida pela própria entidade, e uma chave pública que deve ser acessível a todas as outras entidades. Cada par de chaves partilha uma relação matemática que permite que uma mensagem encriptada com uma das chaves apenas seja desencriptada com a utilização da outra chave do par.

Uma entidade emissora encripta uma mensagem com a chave pública do recetor. Após o envio, a entidade recetora recebe a mensagem encriptada e utiliza a sua chave privada para a desencriptar. A técnica é representada graficamente na figura 2.2.

Por outro lado, se o emissor enviar uma mensagem encriptada com a sua chave privada, a entidade recetora pode ter a certeza que essa mensagem foi enviada pela entidade correta, bastando para isso utilizar a chave pública dessa entidade para desencriptar a mensagem recebida. Este mecanismo representa uma assinatura digital.

Para que tudo funcione em pleno é necessário que as chaves privadas sejam apenas do conhecimento das entidades a elas associadas e que as chaves públicas sejam mapeadas corretamente às entidades correspondentes.

Este tipo de criptografia é um método ideal para o estabelecimento de conexões em ambientes não confiáveis pois, assumindo a possibilidade de partilhar as chaves públicas de um modo fiável, qualquer entidade pode enviar uma mensagem encriptada e ter a certeza que esta apenas é desencriptada com sucesso pela entidade a que se destina [1, cap. *SSL, TLS and Cryptography*].

Os algoritmos utilizados em criptografia assimétrica são complexos refletindo-se na rapidez de processamento. O RSA, baseado na fatorização de grandes números inteiros, é o mais comummente utilizado para este tipo de criptografia.

Figura 2.2: Esquema de criptografia assimétrica

2.1.3 Funções *Hash*

Uma função criptográfica *hash* consiste numa função matemática que a partir de uma entrada de comprimento variável gera uma saída de comprimento fixo, tal como representado na figura 2.3. Estas funções criptográficas obedecem a um conjunto de propriedades, tais como [2]:

- Resistência pré-imagem, i.e. dado um resultado de uma *hash* deve ser computacionalmente muito difícil encontrar o valor da entrada que o gerou;
- Resistência à segunda pré-imagem, i.e. dada uma entrada e o resultado da sua *hash* deve ser difícil encontrar uma outra entrada com o mesmo valor de *hash*;
- Resistência a colisões, i.e. deve ser difícil encontrar duas entradas com o mesmo valor de *hash*.

Este tipo de funções são normalmente utilizadas como um meio de acrescentar eficiência aos métodos criptográficos responsáveis por providenciar integridade e autenticação às comunicações, como por exemplo numa assinatura digital.

Atualmente, os principais algoritmos *hash* são SHA-1 e as variantes do SHA-2.

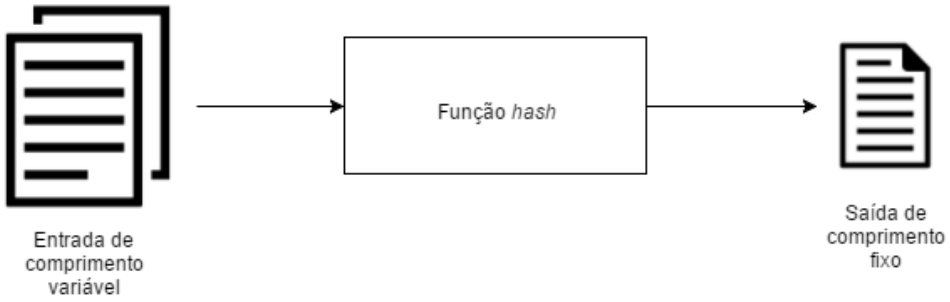


Figura 2.3: Esquema de uma função *hash*

2.1.4 Assinatura Digital

Assinaturas digitais são mecanismos que permitem a verificação da autenticidade e integridade de mensagens digitais. Tal como a sua representação no mundo não digital, estas assinaturas são utilizadas para conferir validade a um determinado conteúdo. Uma assinatura consiste no resultado da encriptação, com a chave privada do autor, da mensagem a transmitir.

Existem vários algoritmos de assinaturas digitais com diferentes formas de operar. Na forma mais utilizada, o emissor aplica uma função *hash* à mensagem que pretende transmitir e utiliza a sua chave privada para encriptar a sua saída, resultando na assinatura digital. Por fim, a entidade emissora envia a mensagem acoplada com a assinatura digital. Este processo encontra-se representado graficamente na figura 2.4.

No que diz respeito à receção, primeiramente o recetor aplica a função *hash* à mensagem recebida e em seguida decifra com a chave pública do emissor o bloco de dados referente à assinatura digital. As saídas referentes aos dois procedimentos anteriores são comparadas e se apresentarem valores idênticos a assinatura é considerada válida. Qualquer alteração do conteúdo da mensagem ou da assinatura dará um resultado negativo na comparação e a consequente rejeição da mensagem recebida. Este processo encontra-se representado graficamente na figura 2.5.

O facto das assinaturas digitais serem assinadas com a chave privada provoca que seja exequível identificar inequivocamente o emissor de uma determinada mensagem e dessa forma garantir o não-repúdio dessa mensagem.

Entre os algoritmos de assinaturas digitais destacam-se o RSA e o ECDSA (*Elliptic Curve Digital Signature Algorithm*).

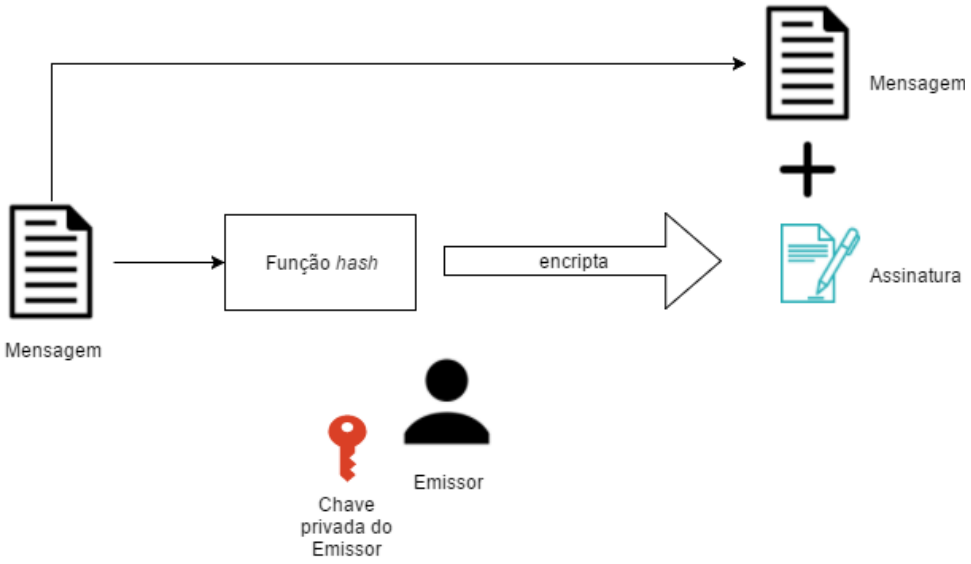


Figura 2.4: Assinatura digital RSA - emissor

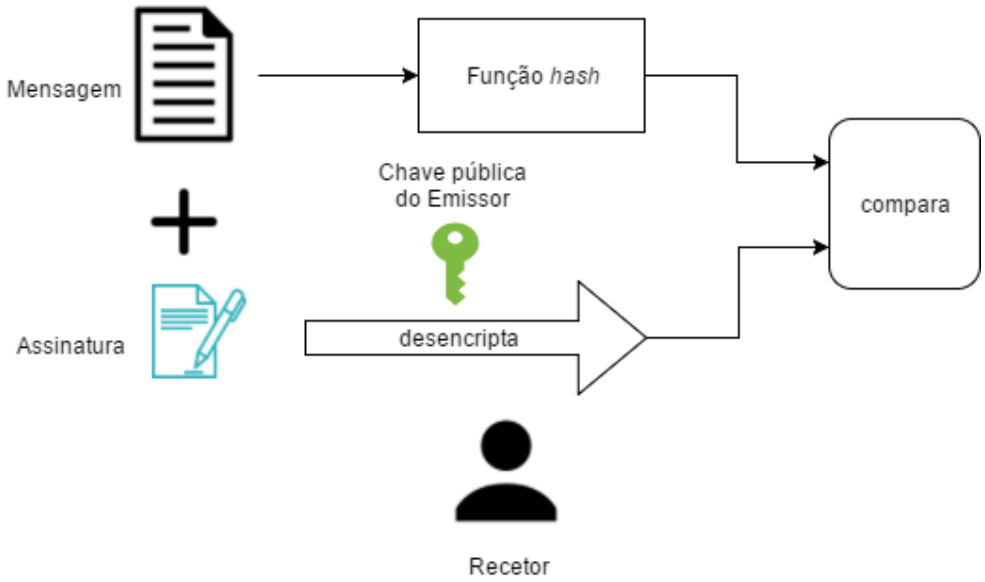


Figura 2.5: Assinatura digital RSA - recetor

2.2 Infraestrutura de Chaves Públicas

Uma infraestrutura de chaves públicas consiste num conjunto de ferramentas e técnicas que, operando cooperativamente, providenciam serviços de gestão de chaves criptográficas. Este tipo de modelo utiliza várias técnicas criptográficas para conceder confidencialidade, autenticação, integridade e não-repúdio às comunicações que nele se suportam. Seguidamente são aclarados os conceitos destas propriedades.

- Autenticação: capacidade de identificar fidedignamente a entidade com quem se estabelece uma comunicação;
- Confidencialidade: garantia de que uma mensagem adequadamente tratada apenas pode ser lida pela entidade a quem foi enviada;
- Integridade: garantia que a informação de uma mensagem não é alterada por entidades não autorizadas sem que as entidades autorizadas se apercebam;
- Não-repúdio: as partes integrantes do processo de comunicação não podem negar o seu envolvimento numa transação.

Um sistema que concretize este conjunto de propriedades é capaz de providenciar comunicações seguras entre as entidades que o utilizam.

Nesta secção são examinados os principais tópicos sobre o tema de infraestrutura de chaves públicas através da análise da sua arquitetura, do seu funcionamento e dos seus principais componentes.

2.2.1 Arquitetura

Uma infraestrutura de chaves públicas tem como base a utilização da criptografia de chaves assimétricas em que, como explicado anteriormente, todas as entidades comunicadoras possuem um par de chaves único constituído por uma chave privada e uma chave pública. O maior dilema deste tipo de técnica criptográfica prende-se na dificuldade em garantir que uma chave pública pertence de facto a uma determinada entidade.

Em meios não confiáveis existe a possibilidade de ataques de interceção (*Man In The Middle, MITM*) onde um atacante, de forma abusiva e oculta, se insere de permeio nas comunicações de outras entidades. Estes ataques tornam-se possíveis quando não existe uma associação legítima entre uma entidade e a sua chave pública.

O modelo de infraestrutura de chaves públicas correntemente aceite, resolve o supracitado problema por intermédio das Entidades Certificadoras que mapeiam uma chave pública a uma entidade através de um documento assinado digitalmente.

Resultado de uma combinação de ferramentas e técnicas, uma infraestrutura corrente de chaves públicas é constituída por um conjunto específico de componentes [3, cap. *Public Key Infrastructure Basics*]:

- **Entidades Certificadoras**

  Definem-se como as entidades confiáveis do sistema e têm como principais funções emitir certificados digitais e gerar informação sobre a revogação dos certificados que deixaram de ser válidos.

- **Solicitadores de Certificados**

  Entidades que solicitam um certificado digital assinado por uma CA.

- **Autoridades de Registo**

  Entidades opcionais que agem como intermediário entre as CAs e os solicitadores de certificados. As suas principais funções são:

  - Receber as solicitações de emissão de certificados;
  - Validar os dados das entidades solicitadoras;
  - Enviar os pedidos para a CA;
  - Entregar o certificado à entidade que o solicitou.

- **Certificados Digitais**

  Documentos eletrónicos assinados digitalmente que associam uma entidade à sua chave pública. Contêm informações sobre a chave pública, a entidade solicitadora e do próprio certificado.

- **Entidades Finais**

  Tratam-se dos utilizadores ou aplicações que utilizam certificados digitais para adquirir a chave pública associada à entidade com qual pretendem comunicar.

- **Repositórios**

  Servidores onde são armazenados os dados relativos aos certificados digitais.

Muitas vezes, a infraestrutura global é simplificada prescindindo das autoridades de registo e até dos repositórios sendo as suas funções delegadas às CAs.

A figura 2.6 representa o relacionamento existente entre os diversos componentes do modelo da infraestrutura de chaves públicas.

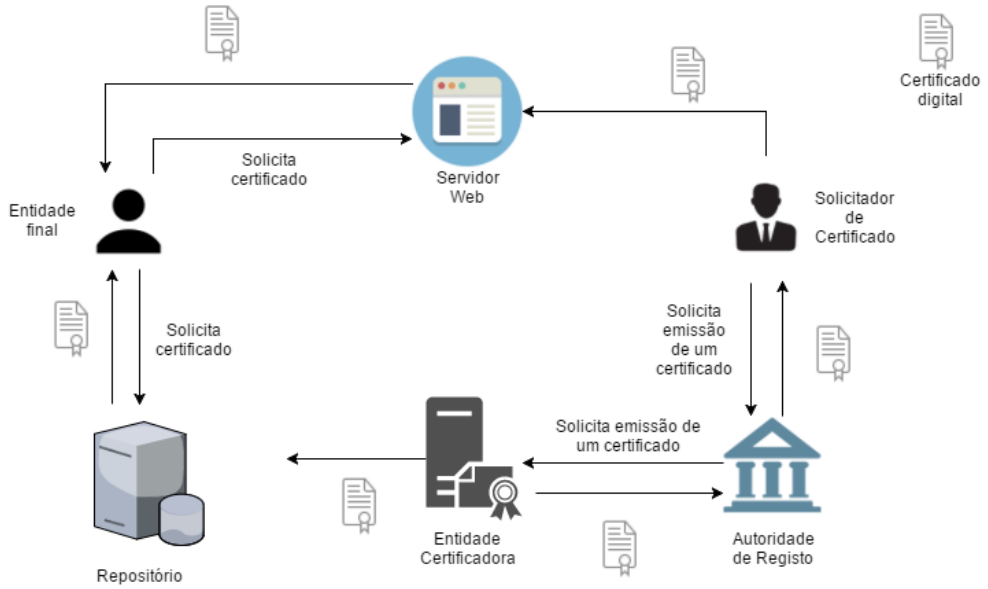


Figura 2.6: Arquitetura do modelo de infraestrutura de chaves públicas utilizado na Internet

2.2.2 Entidades Certificadoras

As Entidades Certificadoras definem-se como as principais entidades confiáveis do modelo e têm como principal finalidade emitir certificados digitais que comprovam a associação de uma entidade à sua chave pública. Todo o modelo de infraestrutura de chaves públicas depende da confiança que os utilizadores têm nestas entidades. Um utilizador ao confiar numa CA confia automaticamente em todos os certificados emitidos por ela.

Para além da emissão de certificados, as CAs são responsáveis por realizar outras funções, nomeadamente a geração de informação sobre certificados revogados, estabelecimento de políticas de segurança e a regência dessas mesmas políticas.

Estas entidades dependem inteiramente da confiança que lhes é conferida e por isso necessitam de conceber políticas que garantam segurança de uma forma transversal. Dessa forma espera-se que realizem um conjunto de ações que previnam o comprometimento dos seus serviços, tais como investigações apropriadas às entidades solicitadoras de certificados, proteção das suas infraestruturas de possíveis intrusos, realização de auditorias, entre outras.

A sua arquitetura consiste num modelo hierárquico onde reinam as CAs raiz. Estas entidades são a base de todas as outras e contam com os seus próprios certificados digitais pré-instalados nas principais aplicações que utilizam este tipo de serviço. Uma CA não-raiz trata-se de uma entidade subordinada a uma CA raiz. Esta relação hierárquica é estabelecida quando uma CA raiz emite um certificado digital para que uma outra entidade possa emitir certificados. Também é possível que uma CA não-raiz emita certificados para uma outra CA não-raiz criando assim uma estrutura hierárquica. Na figura 2.7 é representada uma possível estrutura hierárquica de uma CA raiz.

Os certificados emitidos por CA intermediárias possuem um caminho de certificados que permite à entidade final identificar a CA raiz responsável pelo certificado. Na prática, se um cer-

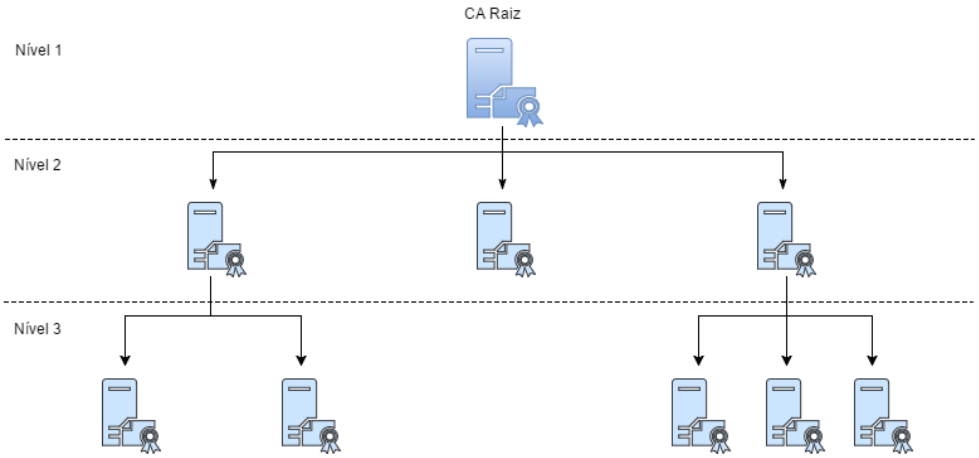


Figura 2.7: Possível modelo hierárquico de uma CA raiz

tificado não foi emitido por uma CA considerada confiável, é realizada uma investigação à sua estrutura hierárquica de modo a derivar se este é da responsabilidade de alguma entidade que seja da confiança da entidade final. Se no caminho de certificados não existir uma referência a uma entidade válida, o certificado é considerado como não fiável pela entidade final.

A figura 2.8 ilustra o caminho de certificados presente num certificado digital, onde é possível verificar a existência de três certificados ligados entre si. O certificado em destaque corresponde ao certificado da página *web* visitada, *www.google.com*. O certificado da entidade raiz encontra-se representado no topo do caminho, correspondendo à entidade *GeoTrust Global CA*. O certificado restante trata-se do certificado da CA intermediária.

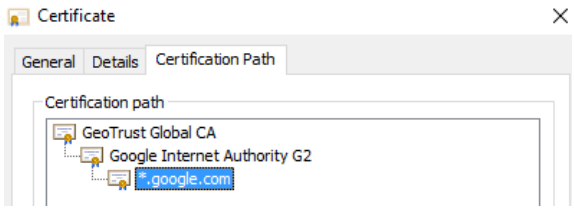


Figura 2.8: Caminho de certificados num certificado digital

Ainda em relação ao exemplo anterior e de modo a evidenciar a diferenciação entre os certificados, na figura 2.9 encontra-se representada a informação, contida nos certificados, referente ao processo de emissão. Nela, constata-se que o certificado referente à página *web* (lado esquerdo) foi emitido pela CA intermediária *Google Internet Authority G2* para o domínio **.google.com*. Seguidamente verifica-se que o certificado referente à entidade intermediária foi emitido pela entidade *GeoTrust Global CA* para a entidade intermediária *Google Internet Authority G2*. Por último, constata-se que o certificado correspondente à entidade raiz foi emitido pela entidade *GeoTrust Global CA* para si própria (certificado auto-assinado).

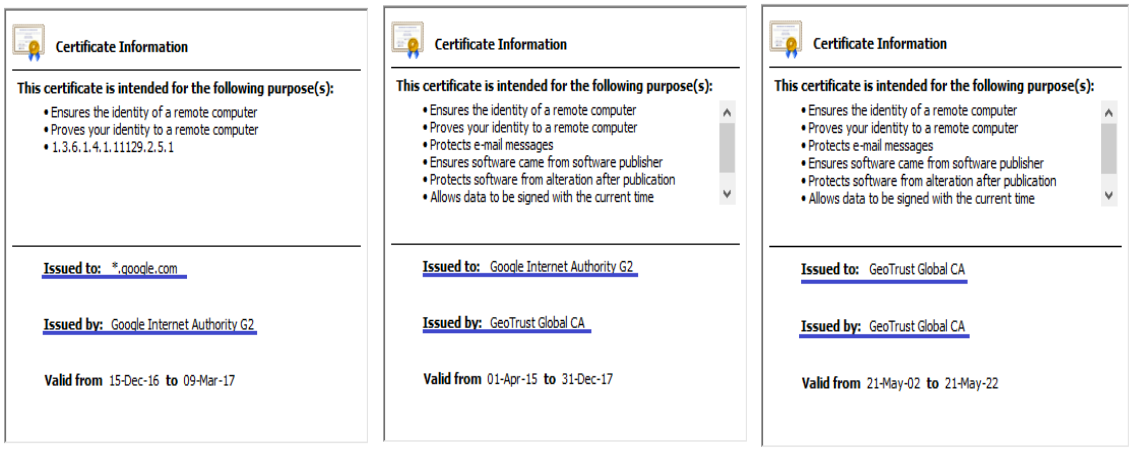


Figura 2.9: Informação sobre a emissão em certificados digitais

Na tabela 2.1 apresenta-se, por uma questão de curiosidade, a distribuição da quota de mercado das principais CAs para a certificação de páginas *web* em 2016 [4].

Tabela 2.1: Quota de mercado das principais CAs

Entidade Certificadora	**Quota**
Comodo	41%
Symantec Group	24%
GoDaddy Group	11%
GlobalSign	8,2%
IdenTrust	7,7%
DigiCert	2,9%
StartCom	2,4%
Entrust	0,6%
Trustwave	0,4%
Verizone	0,4%

2.2.3 Certificados Digitais

Um certificado digital permite vincular uma entidade à sua chave pública através de um documento assinado com a chave privada de uma entidade confiável. Este tipo de documento eletrónico é normalmente emitido por instituições reputadas, Entidades Certificadoras, e contém o conjunto de informações necessárias para o estabelecimento de um canal de segurança entre as entidades comunicantes.

2.2.3.1 Formato X.509

Perante a importância dos certificados digitais tornou-se indispensável utilizar um formato padrão reconhecido globalmente. Atualmente o X.509 consiste no formato de certificado digitais mais utilizado e é aplicado por diversas tecnologias, tais como S/MIME (*Secure/Multipurpose*

Internet Mail Extensions), IPsec (*Internet Protocol Security*), SSH (*Secure Shell*), VPNs (*Virtual Private Networks*), no acesso a páginas *web* SSL, etc.

Um certificado do tipo X.509 tem o seguinte formato [5, cap. *Key Management and Distribution*]:

- **Versão do certificado**

  Este campo atua como um identificador da versão do certificado. Existem três versões deste tipo de formato, das quais a versão x509 v3 é a mais utilizada atualmente.

- **Número de série**

  Identificador único do certificado criado pela entidade emissora.

- **Algoritmo de assinatura do Emissor**

  Especificação do algoritmo utilizado pelo emissor para assinar o certificado.

- **Emissor**

  Identificador da entidade que assinou e emitiu o certificado.

- **Período de validade**

  Intervalo de tempo correspondente ao período em que o certificado é válido. Consiste em duas datas: data de início e data de fim.

- **Entidade Titular**

  Identificador da entidade à qual é associada a chave pública.

- **Informação da chave pública da Entidade Solicitadora**

  Contém a chave pública da entidade para quem é emitido o certificado, a identificação do algoritmo e parâmetros opcionais.

- **Identificador único do Emissor**

  Campo opcional que especifica um identificador único para o emissor. Similar ao campo de identificação do emissor com a diferença de que no caso presente o identificador é único.

- **Identificador único da Entidade Titular**

  Campo opcional que especifica um identificador único para a entidade à qual é associada a chave pública. Similar ao campo de identificação da entidade solicitadora com a diferença de que no caso presente o identificador é único.

- **Extensões**

  Campo utilizado para especificar métodos de associação de atributos a entidades ou a chaves públicas e gestão de interação entre CAs [6].

- **Assinatura**

  Campo que contém a assinatura que valida o certificado digital. Consiste numa *hash* de todos os outros campos assinada com a chave privada da entidade emissora.

2.2.3.2 Revogação de Certificados

A validade dos certificados é crítica para que haja segurança na troca de comunicações. A revogação de um certificado trata-se da anulação da sua validade devido a um comprometimento da legitimidade do mesmo. Algumas das principais razões são:

- Certificação indevida;
- Comprometimento da entidade emissora;
- A entidade que o solicitou deixou de ser válida.

As Entidades Certificadoras são as entidades responsáveis pelas revogações. Existem dois métodos principais para lidar com a revogação de certificados:

- Listas de Revogação de Certificados;
- Online Certificate Status Protocol (OCSP).

Listas de Revogação de Certificados

As listas de revogação de certificados consistem em documentos criados e publicados periodicamente pelas CAs que especificam os certificados que foram revogados antes do término da sua data de validade. As listas publicadas possuem um carimbo com a data da sua emissão e uma assinatura realizada pela CA responsável. Qualquer registo de uma lista inclui o número de série do certificado revogado e a sua data de revogação.

Os utilizadores descarregam a lista atualizada do servidor responsável pelo seu armazenamento e analisam o documento de modo a determinar a validade de um certificado.

Este método tem limitações referentes ao nível de desempenho e de segurança. Quanto ao nível de desempenho, o facto de as listas serem normalmente de grande dimensão torna os acessos e as verificações mais lentas. Quanto ao nível de segurança, o intervalo de tempo entre publicações pode corresponder a um período de possível aceitação de certificados não válidos.

Na figura 2.10 é possível verificar o apontador para uma lista de certificados revogados num certificado SSL. Estas informações constam no campo de extensões de um certificado X.509v3.

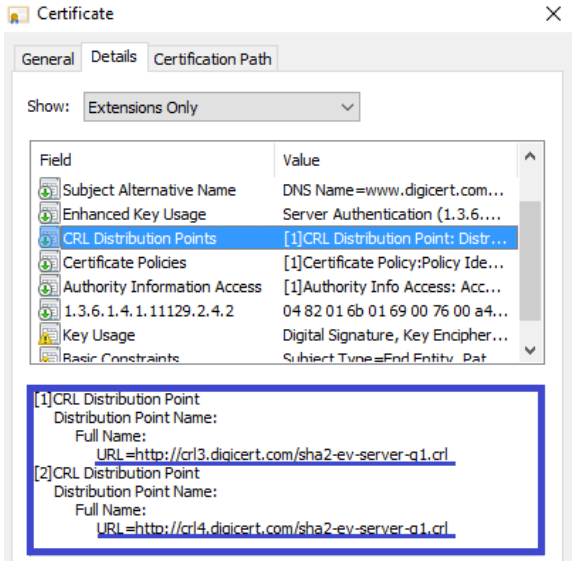


Figura 2.10: Apontador para uma lista de revogação de certificados num certificado SSL X509.v3

Online Certificate Status Protocol (OCSP)

O protocolo consiste na consulta a um servidor (OCSP *responder*) de modo a obter a informação em tempo real sobre o estado de um determinado certificado digital.

Os utilizadores executam simplesmente uma consulta ao servidor OSCP utilizando o número de série do certificado que pretendem validar. O servidor por sua vez responde indicando o estado associado ao certificado. Cada resposta é assinada pelo servidor de modo a assegurar a autenticidade da informação prestada.

Este tipo de serviço não necessita de executar descarregamentos periódicos nem de realizar análises completas de listas porém a sua utilização alberga as limitações de atraso inerentes aos servidores de resposta em tempo real e cria novos pontos de falha à infraestrutura.

Na figura 2.11 é possível verificar o apontador para um servidor OCSP num certificado SSL. Estas informações constam no campo de extensões de um certificado X.509v3.

Uma outra versão OCSP consiste no OSCP *Stapling* que permite que seja o próprio servidor TLS a providenciar a informação do estado de revogação do certificado, eliminando a necessidade de acesso por parte dos utilizadores a servidores externos à comunicação. O funcionamento do método consiste primeiramente na requisição periódica, por parte dos domínios, do estado do seu certificado ao servidor OCSP da CA responsável. A resposta a essas requisições, assinadas digitalmente pelo servidor OCSP, contêm o estado do certificado digital e o *timestamp* correspondente ao momento da requisição. Quando um utilizador acede a um servidor, descarrega o certificado do domínio juntamente com a informação do estado de revogação assinada pelo servidor OCSP. Desta forma o utilizador, através do seu navegador *web*, realiza as devidas verificações ao certificado sem a necessidade de um acesso exterior à comunicação existente[7].

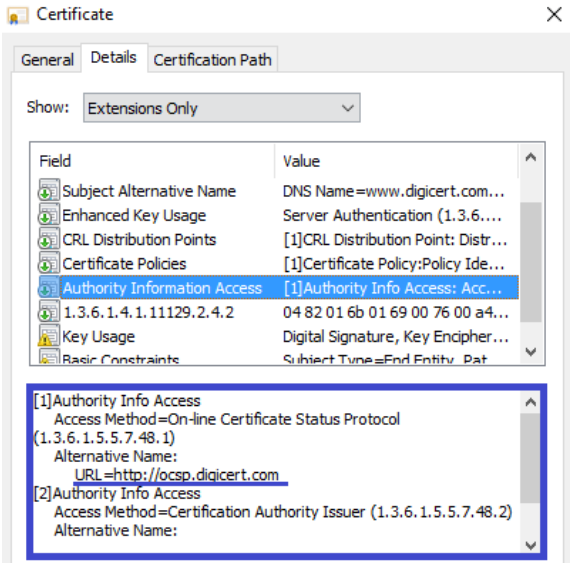


Figura 2.11: Apontador para um servidor OCSP num certificado SSL X509.v3

2.3 Otimizações/Alternativas ao sistema de Entidades Certificadoras

A controvérsia existente em torno do papel das Entidades Certificadoras na infraestrutura de chaves públicas, detalhada no capítulo 3, levou à formação de tecnologias alternativas capazes de substituir ou otimizar o sistema instalado. Segue-se uma breve descrição de algumas das alternativas existentes.

2.3.1 DANE

DANE representa uma tecnologia que visa vincular os certificados digitais utilizados na infraestrutura de chaves públicas ao sistema DNS (*Domain Name System*) utilizando o serviço DNSSEC (*Domain Name System Security Extensions*)[8].

Para compreender as suas particularidades é necessário conhecer os conceitos de DNS e DNSSEC.

DNS

O sistema DNS trata-se de um serviço que efetua a resolução de nomes de domínios em endereços IP (*Internet Protocol*) (e vice-versa), estabelecendo uma associação entre estes, de modo a facilitar os acessos dos utilizadores aos domínios em questão. O sistema é composto por uma estrutura hierárquica de servidores que armazenam a informação de uma forma distribuída. Um domínio é identificado através de um conjunto de nomes separados por pontos e organizados de acordo com a hierarquia existente no sistema [9].

Na estrutura hierárquica do serviço DNS, os servidores situados no topo denominam-se servidores raiz e são responsáveis pelos registos situados na zona raiz DNS e pela indicação, em

resposta a consultas, de servidores de domínios de topo, TLDs (*Top Level Domain*). Um servidor domínio de topo DNS armazena informação acerca dos servidores da camada abaixo da hierarquia (2ª camada) e são identificados através da última secção do nome do domínio (*exemplo.com*). Estas relações podem ser verificadas na figura 2.12.

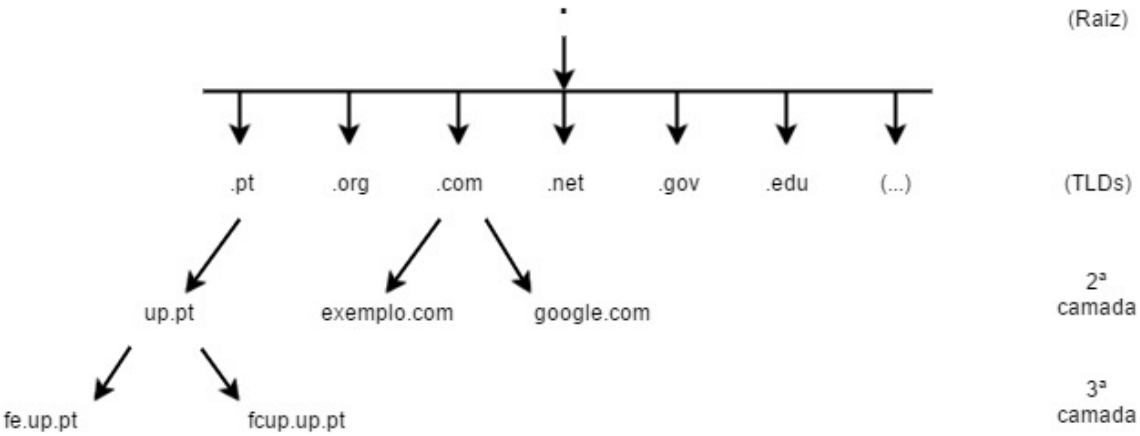


Figura 2.12: Estrutura hierárquica DNS

No que diz respeito ao seu funcionamento normal, assumindo que o sistema operativo do cliente não possui informação em cache, o cliente requisita a resolução de um nome de um determinado domínio no seu endereço de IP a um servidor recursivo DNS. Este tipo de servidores DNS consistem em servidores que não são responsáveis por nenhuma zona mas obtêm informação acerca dos domínios realizando um conjunto de consultas a servidores autoritários. Se o servidor recursivo não tiver em cache a informação requisitada, o processo de resolução continua com o envio de uma consulta a um servidor raiz, ao que este responde indicando um servidor, situado na camada de domínios de topo, que contenha informação sobre o domínio em questão. Seguidamente o servidor recursivo envia a consulta para o servidor indicado no passo anterior, repetindo o processo e descendo pela hierarquia até encontrar o servidor responsável pela informação. A consulta termina com a entrega do endereço de IP ao cliente por parte do servidor recursivo DNS.

A figura 2.13 ilustra a sequência de trocas de informação num exemplo de acesso a um servidor *web* exemplo, *www.exemplo.com*.

DNSSEC

O DNSSEC é uma extensão de segurança do serviço DNS que promove a validação de informação DNS através de assinaturas digitais. Este processo de validação permite que as trocas de informação DNS sejam autenticadas, permitindo uma redução da probabilidade de partilha de informações DNS falsas por parte de entidades mal-intencionadas[10].

Uma das principais ameaças e vulnerabilidades do sistema DNS é a possibilidade da existência de ataques onde a consulta DNS realizada por um cliente é intercetada e respondida por uma entidade mal-intencionada. Este procedimento pode ter como objetivo adulterar a cache do cliente

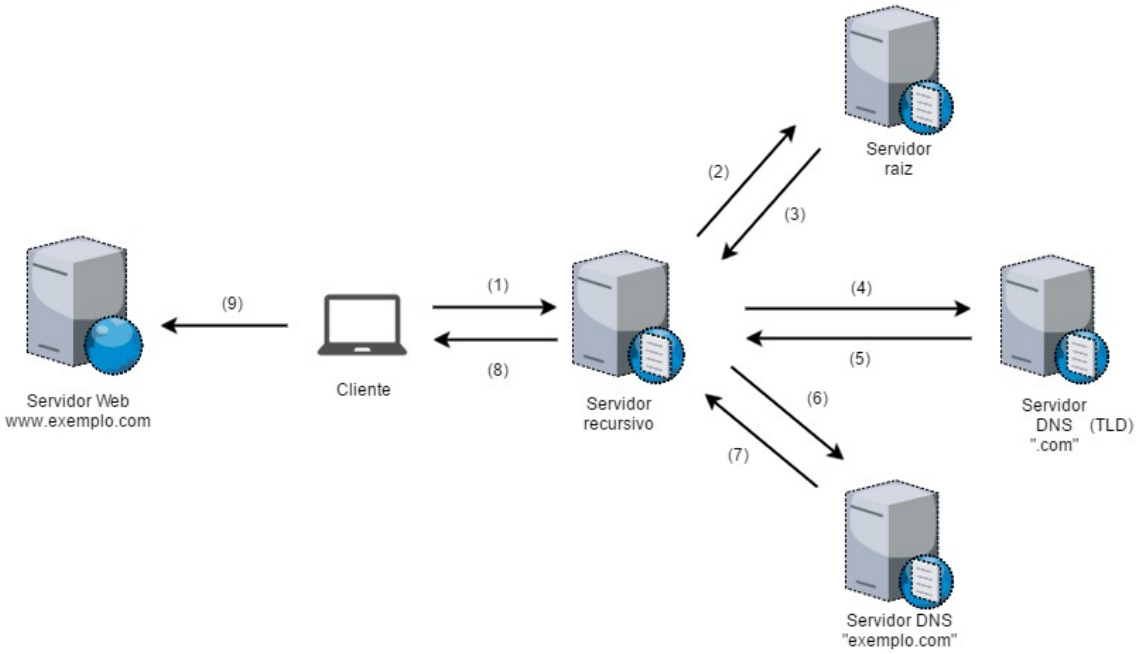


Figura 2.13: Trocas de informação DNS

de modo a que, num acesso ao servidor em questão, este seja direcionado para um outro endereço de IP que não o verdadeiro. Geralmente, este tipo de ataques constituem um meio para realizar outros tipos de ataques (*phishing*, vírus, etc).

O DNSSEC permite, através da criação de novos tipos de registos DNS e da utilização de pares de chaves criptográficas, a utilização de criptografia assimétrica para a validação das informações provenientes dos servidores DNS.

Pares de chaves utilizados:

- KSK: Par de chaves (pública/privada) utilizadas para assinar e verificar chaves de uma zona;
- ZSK: Par de chaves (pública/privada) utilizadas para assinar e verificar registos RRSet (conjunto de registos do mesmo tipo) de uma zona.

O processo tem como base uma cadeia de confiança que se suporta numa imposição na credibilidade da chave da raiz do sistema, obtida pelo cliente através de um outro método que não o DNS. Esta relação base permite que sejam estabelecidas relações de confiança com os servidores DNS, tornando possível a sua autenticação.

Para a implementação do DNSSEC, os administradores das várias zonas necessitam de realizar um conjunto de operações de criação e assinatura de registos por intermédio dos novos pares de chaves de modo a que, numa operação de consulta de um domínio, todas as mensagens enviadas pelos servidores possam ser autenticadas pelo cliente.

O principal objetivo da tecnologia DANE consiste na disponibilização de certificados digitais através do sistema DNS, utilizando DNSSEC para garantir segurança nas comunicações. Este

método permite a continuação da existência dos serviços prestados pelas CAs, mas também possibilita que os detentores dos domínios assinem os seus próprios certificados de forma legítima.

Assumindo o sistema DNS como confiável, uma entidade detentora de um domínio publica o seu certificado num registo DNS assinado com DNSSEC. Os utilizadores realizam o descarregamento do certificado do domínio a que pretendem aceder e procedem à sua validação com o certificado existente no sistema DNS.

Tal como no sistema de CAs, no sistema DNS a confiança é centralizada em determinadas entidades pré-definidas, que no seu caso consistem nas entidades referentes à zona raiz. Atualmente a zona raiz é administrada pela IANA (*Internet Assigned Numbers Authority*), um departamento da entidade ICANN (*Internet Corporation for Assigned Names and Numbers*) situada nos EUA, sem fins lucrativos, que coordena os identificadores únicos do sistema DNS a nível mundial [11]. A centralização da confiança implica os problemas associados à sua utilização, tais como o excesso de poder das entidades responsáveis e a imposição de uma relação de confiança com essas entidades.

A capacidade em tornar legítima a utilização de certificados auto-assinados torna esta tecnologia atrativa tendo em conta o problema da dependência do sistema atual aos serviços das CAs.

Atualmente, a zona raiz do sistema DNS encontra-se assinada[12] o que permite a efetiva implementação de DNSSEC para as restantes zonas da estrutura, e consequentemente, a possibilidade da utilização da tecnologia DANE.

2.3.2 Perspectives

O contínuo aumento de ataques MITM em comunicações entre clientes e servidores *web* foi o principal fator que motivou um conjunto de investigadores da *Carnegie Mellon CyLab* e os professores Adrian Perrig e Dave Anderson a criar o projeto Perspectives[13].

O projeto adiciona uma camada de proteção ao processo de validação de certificados digitais por intermédio da utilização de autoridades notárias que validam a legitimidade dos certificados através de um método de verificação dos registos históricos dos domínios em questão. Desta forma, a sua implementação providencia uma alternativa à maneira como um certificado digital entregue por um servidor *web* é considerado válido pelos clientes.

Na prática, quando um navegador *web* envia uma mensagem de requisição do certificado digital a um determinado servidor, é enviada em paralelo uma consulta a um conjunto de servidores notários com o intuito de conhecer o certificado do domínio visto da perspetiva dessas entidades. Neste sistema, um certificado recebido numa comunicação TLS apenas é considerado válido se coincidir com o certificado recebido pelos servidores notários. As funções dos servidores notários consistem na monitorização de domínios de forma a construir um histórico de certificados para cada um, e na resposta às consultas realizadas pelos clientes.

A sua implementação, para além de permitir que qualquer entidade seja responsável por um ou mais servidores notários, possibilita aos utilizadores a escolha de quais os servidores a quem confiam a segurança das suas comunicações, estabelecendo assim um modelo de confiança descentralizada de escolha livre[14].

Para além da camada de proteção adicional, uma das principais vantagens do projeto consiste na possibilidade da instalação legítima de certificados auto-assinados por parte dos detentores de domínios, permitindo assim a independência com o sistema de CAs. Este processo torna-se válido devido à capacidade independente de validação dos servidores notários.

2.3.3 Convergence

O sistema Convergence foi apresentado por Moxie Marlinspike durante a palestra *SSL and the Future of Authenticity* na conferência *Black Hat* em 2011 [15]. Esta solução tem como base o projeto Perspectives e consiste numa estratégia ágil, distribuída e segura para substituir as CAs de chaves criptográficas públicas [16].

Tal como no projeto Perspectives, são utilizadas autoridades notárias de modo a validar um certificado digital providenciado por um servidor *web*. De entre um conjunto de diferenças técnicas entre as duas soluções, a principal consiste na possibilidade da utilização de diferentes tipos de *back-ends* para as várias autoridades notárias, definindo um valor de consenso entre elas, ou seja, um utilizador pode definir para cada autoridade uma técnica diferente de validação de certificados digitais, tais como a técnica utilizada pelo Perspectives, o DNSSEC, o sistema atual de CAs, entre outras. Esta técnica permite ao utilizador um controlo total sobre a confiança depositada nas autoridades notárias.

A solução foi primeiramente projetada para superar alguns problemas identificados no projeto Perspectives, tais como a plenitude, a privacidade e a capacidade de resposta.

2.3.4 Sovereign Keys

O projeto Sovereign Keys foi criado pela EFF (*Electronic Frontier Foundation*) e consiste numa proposta que visa providenciar assistência na validação de certificados digitais através de servidores inseridos numa infraestrutura específica[17]. Estes servidores, denominados *timeline*, armazenam informação sobre as chaves e os identificadores dos domínios correspondentes, e têm a particularidade de ser constituídos por uma estrutura *append-only*, onde apenas é possível acrescentar dados, sendo impossível alterar ou eliminar registos.

As entidades detentoras dos domínios, depois de terem na sua posse o certificado digital para o domínio em questão, criam uma chave, denominada soberana, que é enviada para um servidor *timeline* juntamente com o certificado.

Do lado do servidor *timeline*, após a receção de um pedido de anexação, são realizadas um conjunto de verificações à informação recebida para que a chave soberana e o identificador da entidade presente no certificado sejam anexados à estrutura.

Um utilizador, quando acede a um determinado domínio, descarrega o certificado com uma assinatura adicional realizada com a chave soberana. Ao mesmo tempo, requisita a um servidor *timeline* os registos associados ao domínio em questão. De modo a realizar a validação do certificado, o utilizador verifica a assinatura adicional com a chave soberana adquirida através do servidor *timeline*.

Esta abordagem exige a existência de entidades confiáveis no modelo pois é requerida uma identificação legítima do detentor do domínio para que seja possível executar um registo num servidor *timeline*, que também ele, deverá ser confiável.

A sua infraestrutura e técnicas específicas fazem com que esta solução albergue um conjunto de problemas, mais especificamente: a confiança imposta aos utilizadores nos servidores *timeline*, o aumento de pontos de ataques à infraestrutura, os custos de transição referentes à infraestrutura necessária, etc.

2.3.5 Certificate Transparency

A abordagem Certificate Transparency, em português "Transparência de Certificados", é uma iniciativa da Google que visa a deteção de certificados SSL não legítimos através de uma monitorização e auditoria de certificados emitidos pelas CAs. Esta técnica requer a utilização de uma estrutura específica que engloba três tipos de componentes, nomeadamente: servidores de registos (logs), monitores e auditores [18].

Os servidores de registos (*logs*) armazenam certificados numa estrutura *append-only*, onde não é possível editar nem remover registos, e permitem a realização de monitorizações e auditorias aos seus registos por parte de qualquer entidade, possibilitando uma verificação pública da integridade das suas estruturas e a consulta da existência de certificados nos seus registos. Os servidores são compostos por uma estrutura de dados denominada *Merkle Tree* composta por um conjunto de nós, em que a *hash* de cada nó depende de todos os nós a si subsequentes. Esta técnica permite realizar verificações de integridade aos servidores de registo de uma forma eficiente.

Os monitores consistem em servidores que periodicamente examinam os servidores de registos de modo a efetuar a comprovação do seu correto funcionamento.

As principais funções dos auditores, usualmente implementados nos navegadores *web*, consistem na realização de consultas aos servidores de registos de modo a averiguar a existência de um determinado certificado digital na sua estrutura, e a execução de verificações à integridade dos servidores de registos.

Na prática, após o processo de emissão, os certificados são enviados para um servidor de registos. Após a receção de um certificado, o servidor de registos responde com um SCT (*Signed Certificate Timestamp*) que consiste numa promessa de inclusão do certificado à sua estrutura. Por último, no processo de estabelecimento de comunicação entre uma entidade final e um servidor seguro, o navegador *web* necessita de, para além realizar a normal validação do certificado, validar o SCT providenciado pelo servidor *web* de modo a tomar como legítimo o acesso ao domínio.

Através desta técnica, qualquer entidade pode inspecionar a emissão de novos certificados através da consulta de um conjunto de servidores públicos. Uma das contrariedades desta abordagem é a não admissão de certificados auto-assinados, dado existir a necessidade da utilização de CAs no estabelecimento de relações de confiança.

2.3.6 HTTP Public Key Pinning

HPKP (*HTTP Public Key Pinning*) é uma abordagem que visa evitar a admissão, por parte das entidades finais, de certificados SSL emitidos de forma ilícita. Esta camada de segurança adicional é obtida através de um conjunto de informações específicas contidas num bloco de dados, *header*, enviado pelos servidores *web* via protocolo HTTP [19].

O atual sistema de infraestrutura de chaves públicas possui a debilidade de permitir que as CAs emitam certificados digitais para um domínio sem a sua prévia autorização, o que possibilita a emissão de certificados não legítimos para qualquer domínio *web* se a infraestrutura de uma determinada CA for comprometida.

A otimização HPKP possibilita que os domínios *web* informem os utilizadores finais de um conjunto de entidades autorizadas, permitindo que os utilizadores finais validem as entidades presentes no certificado descarregado através da comparação com o conjunto de entidades informadas pelo domínio. Na prática, as entidades finais recebem um *header* via HTTP que contém um conjunto de *pins* referentes às chaves públicas das entidades válidas associadas ao domínio. Os *pins* correspondem a resultados de algoritmos de *hash* (*fingerprints*) codificados, referentes ao campo *Subject Public Key Info* de certificados X.503, normalmente pertencentes à cadeia de certificados do domínio em questão. Os navegadores *web*, por sua vez, armazenam a informação presente no *header* para que, num próximo acesso, para além de efetuar as verificações referentes à identificação de uma CA confiável na cadeia de certificados, procedam às validações subjacentes à técnica de HPKP.

Segue-se um exemplo de um *header* HPKP, onde numa primeira instância, é possível identificar o período de validade, expresso em segundos, das informações presentes no bloco de dados. Verifica-se também a existência de dois *pins* referentes à identificação *Subject Public Key Info* das entidades responsáveis, que consistem na codificação em base 64 das *fingerprints* obtidas através do algoritmo de *hash* SHA-256. A abordagem HPKP impõe a indicação de pelo menos dois pins, em que um deles deverá estar contido na cadeia de certificados referente ao domínio e o outro deverá consistir num *pin* de reserva, recomendavelmente associado a um certificado de outra CA, de modo a prevenir possíveis erros nas configurações. A última linha do exemplo apresenta um apontador para um domínio responsável por receber informações sobre validações de *pins* falhadas [1, cap. *HSTS, CSP, and Pinning*].

```
Public-Key-Pins:  max-age=7200;
pin-sha256="YTQxNjBiODAwMzg1MGM3MjRhM2FlO2e1xGJhNWIwM2U= ";
pin-sha256="NjI0OGI3NI4OTBNSWUxYTYzMzVjZTNhY1WJiYmU2NjU=”;
report-uri=HTTP://pkpinning-report.com/example
```

Com esta técnica é possível diminuir as probabilidades da ocorrência de ataques MITM devido ao procedimento de validação da identificação dos domínios através de um conjunto de informações previamente estabelecidas. Esta abordagem tem como principais desvantagens a implementação baseada em TOFU (*Trust on First Use*), ou seja, na imposição da confiança no primeiro

acesso a um domínio, e o fato de serem necessários conhecimentos específicos por parte dos detentores dos domínios de modo a realizar uma configuração apropriada da técnica. Atualmente os navegadores *web Chrome*, *Firefox* e *Opera* são compatíveis com o HTTP Public Key Pinning [20].

2.3.7 TACK

O TACK (*Trust Assertions for Certificate Keys*) trata-se de uma proposta de *pinning* de chaves públicas idealizada por M. Marlinspike e T. Perrin que tem como principal objetivo eliminar, no sistema atual de distribuição de chaves públicas, a vulnerabilidade referente à possibilidade de que qualquer entidade certificadora tem em emitir um certificado digital aparentemente válido para um domínio, sem a devida autorização. Esta vulnerabilidade torna-se crítica pois proporciona um ponto de falha capaz de ser explorado por parte de atacantes para realizar ataques MITM [21].

A ideia principal consiste na possibilidade de os utilizadores realizarem o *pin* de uma chave TSK (*Tack signing key*) escolhida pelo servidor ao qual pretendem comunicar, que assina a chave pública do servidor. Um utilizador, ao realizar um acesso a um servidor, requisitará o TACK inserido na comunicação TLS. O TACK corresponde à chave pública TSK e a assinatura da chave pública do servidor, realizada através do par TSK. Quando o utilizador se deparar uma segunda vez com a mesma combinação domínio-TSK ativará o pin entre o domínio e o TSK durante um período igual ao período de tempo que a combinação foi observada.

Desta forma, a partir do momento da criação do *pin* até ao término do seu período de validade, sempre que o utilizador aceda ao servidor em questão irá requerer que todas as chaves públicas recebidas sejam assinadas pela TSK armazenada.

A utilização de um par de chaves TSK criado pelo detentor de domínio permite que domínios com mais do que um servidor sejam capazes de utilizar a mesma chave TSK para prover mecanismos de *pinning* aos seus utilizadores, erradicando o problema de inflexibilidade de outras soluções de *pinning*. O período de validade dos *pins* limita os erros provocados por más configurações e outros tipos de conflitos.

Concluindo, a utilização da proposta TACK, que consiste numa solução que opera através de uma perspetiva diferente das soluções de *pinning* anteriores, contribui para uma diminuição da probabilidade da existência de ataques MITM.

2.4 Modelos Alternativos

2.4.1 PGP

O PGP (*Pretty Good Privacy*) trata-se de um sistema criptográfico que utiliza uma combinação de algoritmos criptográficos simétricos e assimétricos de modo a providenciar confidencialidade e autenticidade às comunicações suportadas por si [22].

O sistema PGP estabelece legitimidade na associação de uma entidade à sua chave criptográfica pública através de um modelo de confiança descentralizado, contrastando com a solução implementada pela infraestrutura de chave públicas clássica.

Na sua abordagem, cada utilizador possui uma *keyring* que consiste numa coleção de chaves públicas correspondentes às entidades com quem o utilizador pretende comunicar. A principal particularidade do sistema consiste na possibilidade de qualquer utilizador poder assinar a chave pública de uma outra entidade, tornando-se um introdutor dessa chave. Se um utilizador considerar uma determinada entidade como um introdutor válido, as chaves por ele assinadas serão classificadas com um certo grau de validade na sua *keyring*, sendo que o grau de validade da classificação depende da confiança imposta pelo utilizador na entidade introdutora [3, cap. *Understanding PGP*]. Esta técnica, replicada pelos diferentes utilizadores, permite a constituição de uma rede de confiança (*web of trust*).

O método criptográfico utilizado pelo PGP para providenciar confidencialidade consiste, numa primeira fase, na criação de uma chave de sessão aleatória que é utilizada para a encriptação da mensagem. Seguidamente, a chave de sessão é encriptada com a chave pública da entidade recetora e enviada juntamente com a mensagem encriptada. Na receção, o recetor primeiramente utiliza a sua chave privada para desencriptar a chave de sessão e depois utiliza a chave de sessão para desencriptar a mensagem encriptada.

O método criptográfico para providenciar autenticidade é idêntico ao descrito na secção 2.1.4 referente ao tema de assinaturas digitais.

2.4.2 SPKI

O SPKI, *Simple Public-Key Infrastructure*, consiste numa proposta de padrão de criptografia de chaves públicas que providencia resolução de nomes e serviços de autorização sem a dependência de um modelo centralizado [23].

A atribuição de nomes globais a entidades é válida quando se tratam de sistemas de dimensão limitada, onde a identificação de uma determinada entidade consiste num processo simples, dado o carácter único e restrito da gama de utilizadores. A utilização de nomes globais em sistemas também globais, tal como a Internet, implica problemas na identificação das inúmeras entidades, sendo um processo contraditório à perceção humana de nomes.

A abordagem SPKI faz uso da implementação de nomes locais pertencentes a um *namespace*, identificado por uma chave, sem a necessidade de entidades terceiras que assegurem mapeamentos válidos. Dentro de um *namespace* é possível definir nomes básicos, que consistem numa chave pública e um identificador da entidade, e nomes compostos que consistem numa chave pública seguida de dois ou mais identificadores.

O sistema engloba dois tipos de certificados: os certificados de nomes e os certificados de autorização. Quanto aos certificados de nomes, estes providenciam uma definição de um nome local no *namespace* do emissor. Este tipo de certificados contêm quatro atributos (tuplos):

- Emissor: Chave pública. Identificador da entidade que está a definir o nome no seu *namespace*;
- Nome;

- Entidade: Identificador do nome;
- Datas de validade: Data "não antes" e data "não depois".

O SPKI prevê ainda a utilização de certificados de autorização, que têm como principais objetivos a vinculação e delegação de atributos de autorização a uma determinada entidade. Num certificado de autorização, para além dos campos referentes ao emissor, ao visado, às datas de validade e à autorização em questão, existe um campo de delegação que indica se a entidade visada tem ou não permissão para delegar a autorização contida no certificado a outras entidades.

2.4.3 Blockchain

O conceito de Blockchain tornou-se popular como o suporte principal do sistema de moeda da Internet *bitcoin*[24]. Esta tecnologia consiste numa base de dados transacional que armazena um conjunto de informação proveniente dos nós da rede à qual se encontra inserida. De uma forma simples, trata-se um sistema descentralizado, no qual os nós participantes partilham uma *ledger* (conjunto de registos), em que cada registo da *ledger* contém uma associação ao registo que o antecede, criando assim uma ligação entre todos os registos existentes.

No funcionamento do *bitcoin*, cada cliente é identificado através da sua chave pública e cada transação consiste numa mensagem assinada com a chave privada da entidade que originou a transferência.

Um bloco de transações consiste num conjunto de recentes transações que ainda não foram integradas na *ledger*. O processo de integração de blocos na *ledger* denomina-se *mining* e é protagonizado pelos nós trabalhadores do sistema (*miners*). Estes nós, para levarem a cabo a inclusão de blocos na *blockchain* necessitam de realizar um processo de computação que consiste numa prova de trabalho que possui a peculiaridade de ser fácil de validar a sua solução mas extremamente difícil de encontrá-la. O primeiro *miner* a encontrar uma solução, tem o direito de incluir o bloco na *ledger* e como prémio obtém uma quantia de bitcoins. Uma vez incluído o novo bloco, é realizada um atualização por parte de todos os nós à *ledger*, de modo a sincronizar toda a informação do sistema. Como cada bloco contém uma ligação ao bloco anterior, é criada uma corrente de blocos (*blockchain*), como já foi referido.

O sistema de *blockchain* pode ser adaptado para um sistema de nomes públicos, criando a possibilidade de alterar o paradigma da utilização de entidades certificadoras para certificar associações de entidades às suas chaves públicas. Existem já várias soluções que exploram este conceito, tais como o "Namecoin" [25], o "Blockstack" [26], etc.

A adaptação de um sistema baseado numa *blockchain* tem como principais vantagens: descentralização e transparência, devido à distribuição dos dados pelos vários nós tornando toda a informação pública; robustez, devido à ausência de um ponto central de falha; integridade, dada a propriedade imutável dos registos que constituem a *blockchain*. As principais desvantagens de um sistema deste género consistem na latência originada pela manutenção de uma cópia exata da *ledger* por parte dos nós integrantes e pelo contínuo aumento do tamanho da *ledger* devido à sua

natureza acumulativa; no pagamento aos *miners* pois, mesmo que o sistema não envolva a transferência de bens monetários, o trabalho realizado pelos *miners* terá de ser valorizado; nos seus custos de transição.

2.5 Considerações

Neste capítulo foi descrito o estado da arte especificando as técnicas base de criptografia, os conceitos principais de uma infraestrutura de chaves públicas e algumas das principais alternativas existentes ao papel das Entidades Certificadoras.

Numa primeira fase foram explicitadas as técnicas basilares de criptografia e aclarado o conceito de infraestrutura de chaves públicas explicitando o seu principal objetivo e os serviços por si prestados. Ainda no ponto de infraestrutura de chaves públicas foi especificada a arquitetura do modelo juntamente com a definição dos seus vários componentes. Devido à sua importância e carácter agregador dos demais componentes foram abordados com mais detalhe os conceitos de CAs e de certificados digitais.

Posteriormente foi apresentada uma descrição resumida de sete relevantes otimizações/alternativas existentes ao sistema de CAs de chaves públicas.

O capítulo culmina com uma breve exposição de três modelos alternativos ao sistema atual de distribuição de chaves criptográficas públicas.

Os temas anteriores são fundamentais para a compreensão e análise dos aspetos retratados nas seguintes etapas do documento.

Capítulo 3

Caracterização do Problema

Neste capítulo realiza-se uma descrição do problema em questão. Para esse efeito, numa primeira instância são enumeradas e explicitadas as principais fragilidades das Entidades Certificadoras de chaves públicas. O capítulo termina com a descrição de dois casos reais de ataques e as suas principais consequências.

3.1 Fragilidades das Entidades Certificadoras

Devido à expansão exponencial da Internet, com o decorrer do tempo foram encontradas fragilidades no modelo de infraestrutura de chaves públicas, nomeadamente nas CAs. Seguem-se algumas das principais fragilidades implícitas à utilização dos serviços providenciados por uma CA.

3.1.1 Comprometimento de uma Entidade Certificadora

Dependendo totalmente da fiabilidade das CAs, os utilizadores estão sujeitos ao possível comprometimento da segurança dessas entidades. Existem vários casos, abordados na secção seguinte, que comprovam a possibilidade real da existência de falhas de segurança e, como resultado, todas as consequências inerentes a essa situação.

A título de exemplo, um ataque com sucesso a uma CA pode permitir que o invasor se apodere da sua chave privada e com ela emitir certificados "não-legítimos" considerados válidos pelos utilizadores. Esses certificados podem mapear uma entidade à chave pública do atacante e assim permitir a possibilidade de ataques MITM.

O comprometimento de uma CA motiva a aplicação, por parte dessa entidade, de políticas de resposta a estas situações de modo a repor as suas funcionalidades e garantir a segurança dos utilizadores finais. Dependendo do tipo de ataque, estes procedimentos de recuperação podem ter consequências penosas para as partes integrantes do sistema.

3.1.2 Confiança Cega

O estabelecimento de confiança em certificados é um processo pouco transparente do ponto de vista do utilizador visto que as principais aplicações pré-instalam uma lista com as CAs que consideram seguras. A partir do momento da sua instalação, essas entidades tornam-se responsáveis por todos os certificados que vão ser aceites pela aplicação.

Resignadas ao corrente funcionamento e com o intuito de simplificar a experiência de utilização dos seus clientes, as aplicações contribuem para um sistema onde os utilizadores têm pouca expressão no que diz respeito à definição dos seus parâmetros de segurança.

Apesar de ser um processo reversível, ou seja, um utilizador tem a capacidade de marcar como não confiável uma entidade pré-instalada e até de acrescentar novas entidades em que confia, a maior parte não possui os conhecimentos para tal, o que se traduz numa confiança nas CAs não criteriosa por parte dos utilizadores finais.

3.1.3 Inflexibilidade

Numa infraestrutura de chaves públicas, um utilizador final não dispõe de um meio termo no que diz respeito à confiança que nutre por uma CA. Um utilizador apenas tem duas opções, ou confia na CA e aceita todos os certificados a ela associados, ou renuncia a entidade e rejeita todos os certificados a ela subordinados.

Uma das principais consequências da inflexibilidade consiste na imposição de conferir um nível de confiança igual a entidades que não providenciam as mesmas garantias.

A título de exemplo, a falta de agilidade do sistema provoca que uma simples rejeição de uma CA raiz origine o possível bloqueio de uma parte das páginas *web* normalmente consultadas ou nas quais se tem interesse, pois os certificados associados a essa entidade e todos os emitidos pelas entidades certificadoras abaixo na hierarquia deixam de ser válidos para esse utilizador. Este problema é particularmente grave quando se tratam de entidades de grandes dimensões que são responsáveis pelos certificados de uma quota considerável da Internet. (Ver lista de quotas da tabela 2.1)

3.1.4 Emissão de Certificados

O facto de as CAs poderem emitir um certificado para um domínio sem o consentimento do próprio detentor do domínio e a falta de transparência das políticas de emissão conferem mais pontos frágeis ao sistema.

A relação de confiança exigida requer que os utilizadores finais confiem plenamente em algumas CAs raiz, o que significa confiar em todas as entidades a elas submissas hierarquicamente. Com a existência de inúmeras CAs aumenta a probabilidade de erros e/ou omissão de políticas de segurança no processo de emissão, traduzindo-se em certificados não legítimos e consequentemente num fator de risco para os utilizadores [27].

3.1.5 Natureza Comercial

Com a evolução da Internet, as CAs de chaves públicas tornaram-se num negócio bastante rentável e como em qualquer operação comercial a procura de lucro representa a principal motivação para o desempenho dos serviços prestados. Esse fator levanta várias questões éticas ao nível do funcionamento dessas entidades, em como a ganância económica pode influenciar medidas que afetem a segurança dos seus utilizadores.

A emissão de certificados sem a devida identificação do solicitador e auditorias de qualidade questionável são alguns exemplos de possíveis atos suspeitos realizados pelas Entidades Certificadoras [1, cap. *Public Key Infrastructure*].

3.2 Exemplos de Casos Reais

O elevado grau de importância das Entidades Certificadoras na infraestrutura de chaves públicas faz com que sejam um alvo predileto de ataques. Ao longo do tempo foram notícia vários ataques a estas entidades que geraram graves consequências. Estes ataques permitiram aumentar o nível de exposição das principais vulnerabilidades do sistema. Seguem-se dois exemplos eleitos pela sua dimensão e destaque.

3.2.1 DigiNotar

A DigiNotar foi uma CA holandesa que se dedicava à emissão de certificados digitais. Em 2011 foi alvo de um ataque informático às suas infraestruturas que resultou num comprometimento total das garantias de segurança providenciadas pela entidade, visto que o invasor conseguiu ter acesso absoluto aos servidores responsáveis pela emissão dos certificados. Durante o ataque foram emitidos um conjunto de certificados para domínios de alta importância assinados com a chave privada da DigiNotar, tornando possível ao atacante a execução de ataques MITM.

Após o incidente, a empresa "FOX-IT" foi incumbida de executar uma auditoria ao ataque de modo a identificar as causas e as suas principais consequências. A auditoria revelou a emissão de pelo menos 531 certificados fraudulentos[28].

Pouco tempo depois do incidente vir a público e apesar de ter cumprido os protocolos de recuperação de desastre, a empresa entrou em falência devido à falta de confiança por parte dos utilizadores.

3.2.2 *Comodo*

A *Comodo* é um grupo de empresas que se dedica ao desenvolvimento de serviços de segurança informática. Um dos ramos da empresa consiste na *Comodo CA* que se dedica à emissão de certificados digitais. No ano de 2011, a *Comodo* admitiu ter sido vítima de um ataque informático que provocou a emissão de certificados digitais fraudulentos. Uma autoridade de registo pertencente à infraestrutura da *Comodo* em Itália foi comprometida permitindo que o invasor conseguisse emitir nove certificados digitais para sete domínios de grande projeção pública [29]:

- login.live.com
- mail.google.com
- google.com
- login.yahoo.com
- login.skype.com
- addons.mozilla.org
- "Global Trustee"

Apesar da gravidade da situação, dada a posição da *Comodo* como a CA com maior quota de mercado, a empresa manteve a sua atividade comercial.

Capítulo 4

Estrutura de Chaves Públicas Interna

Este capítulo, em que se descreve a implementação de uma estrutura de chaves públicas interna, tem como objetivo provar a possibilidade e facilidade da construção de uma infraestrutura de chaves públicas dentro de uma instituição ou corporação, sem a necessidade do envolvimento de entidades certificadoras externas.

Na descrição da implementação são descritos: o *software* utilizado, a rede produzida e os vários passos que foram necessários para a criação de cada componente da estrutura.

Os detalhes, verificações e validações da implementação podem ser consultadas no Anexo A deste documento.

4.1 *Software*

Todo o *software* utilizado no desenvolvimento da infraestrutura de chaves públicas é *open source*. Umas das grandes vantagens deste tipo de *software* consiste na capacidade de possibilitar a alteração do código fonte por parte de qualquer entidade para beneficio próprio.

OpenSSL

OpenSSL[1] é uma biblioteca de ferramentas criptográficas que reúne um conjunto de mecanismos necessários para a implementação dos protocolos SSL (*Secure Socket Layer*) e TLS (*Transport Layer Security*). SSL consiste num protocolo criptográfico que disponibiliza confidencialidade e autenticidade a comunicações através de mecanismos criptográficos. TLS consiste numa versão avançada do protocolo SSL.

Apache

Apache[2] é um *software* que permite a implementação um servidor HTTP. Trata-se de uma das variedades de servidores *web* mais utilizada em todo o mundo.

[1] https://www.openssl.org
[2] https://httpd.apache.org/

Postfix

Postfix[3] é um *software* que implementa um *Mail Transfer Agent*, MTA. Este tipo de *software* é responsável pelo transporte de mensagens de correio eletrónico desde que saem da organização onde se situa o emissor até à organização onde está o recetor (endereçado). As mensagens são passadas entre MTAs através do protocolo SMTP (*Simple Mail Transfer Protocol*). A intermediação entre os utilizadores e os MTAs é realizada através de *software* de interface designado por User Agent, UA.

Bind

Bind[4] é um *software* que possibilita a implementação de um servidor DNS (*Domain Name System*). A versão utilizada consiste na variante *bind9*.

Wireshark

Wireshark[5] é um *software* capaz de analisar o tráfego de uma determinada rede através da inspeção dos vários protocolos de rede utilizados. A sua utilização é conveniente quando é necessário solucionar problemas numa rede, examinar tráfego de um determinado protocolo organizado por ordem temporal, etc. No contexto deste projeto, foi utilizado como ferramenta de depuração da implementação.

Icedove

Icedove[6] é um cliente de correio eletrónico baseado no *Thunderbird*, adotado para a distribuição *Linux Debian*.

4.2 Rede

Na figura 4.1 apresenta-se a estrutura de rede utilizada para a simulação de uma estrutura de chaves públicas numa *intranet*. A rede utilizada foi a 192.168.56.0/24, que possui quatro máquinas conectadas, qualquer uma com acesso a todas as outras. A máquina *Debian Root* consiste numa máquina pertencente à estrutura de chaves públicas, normalmente desligada da rede, estando apenas conectada quando utilizada para assinar certificados de entidades certificadoras.

O nome *My Company* foi o escolhido para nomear a rede local e todos os serviços implementados foram denominados em sua função.

[3] https://help.ubuntu.com/community/Postfix
[4] https://www.isc.org/downloads/bind
[5] https://www.wireshark.org/
[6] https://packages.debian.org/sid/icedove

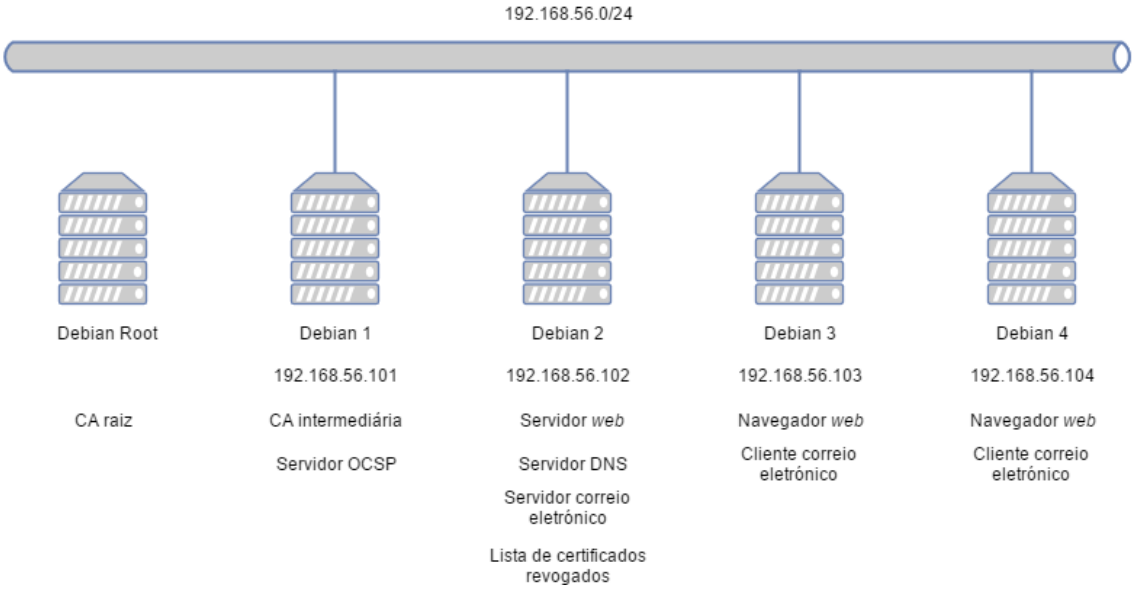


Figura 4.1: Estrutura da rede

Debian Root

Máquina utilizada para alojar a CA raiz do sistema. Esta entidade situa-se no topo da estrutura hierárquica de certificados e por essa razão dispõe de um certificado auto-assinado. Em virtude da premissa de que a CA raiz se trata da entidade confiável do sistema, todos os certificados possuem um elo de subordinação ao certificado raiz com o intuito de prover confiança às entidades que o utilizem. A sua principal função consiste na emissão de certificados de CAs intermediárias. Devido ao papel fulcral na estrutura de chaves públicas, a máquina foi colocada fora da rede (*offline*) de maneira a que esteja totalmente protegida de qualquer tentativa de ataque proveniente da rede.

***Debian 1*: (192.168.56.101)**

A principal função da máquina *Debian 1* consiste no alojamento da CA intermediária da estrutura. Esta entidade é responsável por assinar os certificados dos servidores e clientes do sistema e tem o seu certificado assinado pela CA raiz. Esta máquina serve também como alojamento ao servidor OCSP do sistema, responsável pela resposta a pedidos de clientes acerca do estado de revogação de certificados.

***Debian 2*: (192.168.56.102)**

Máquina utilizada para alojar o servidor *web*, o servidor de correio eletrónico e o servidor DNS da *intranet*. A lista de certificados revogados referente à estrutura de certificados implementada é também disponibilizada através desta máquina.

***Debian 3*: (192.168.56.103) e *Debian 4*: (192.168.56.104)**

As máquinas *Debian 3* e *Debian 4* são utilizadas para simular acessos de utilizadores da rede aos diversos serviços disponibilizados.

Na imagem 4.2 encontra-se representada a estrutura de certificados implementada. Esta organização tem como entidade principal uma CA raiz responsável pela emissão do certificado da entidade certificadora intermediária. A CA intermediária tem o papel de assinar os certificados referentes aos servidores e clientes do sistema.

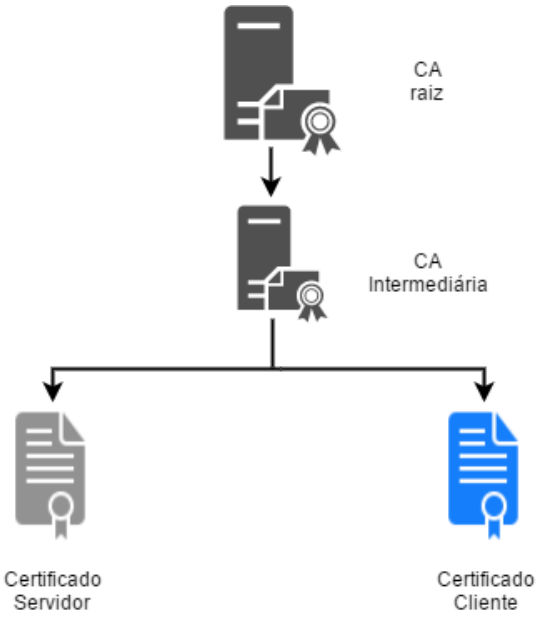


Figura 4.2: Estrutura de certificados

A figura 4.3 ilustra o fluxo de operações associado à implementação dos diversos componentes da estrutura concretizada.

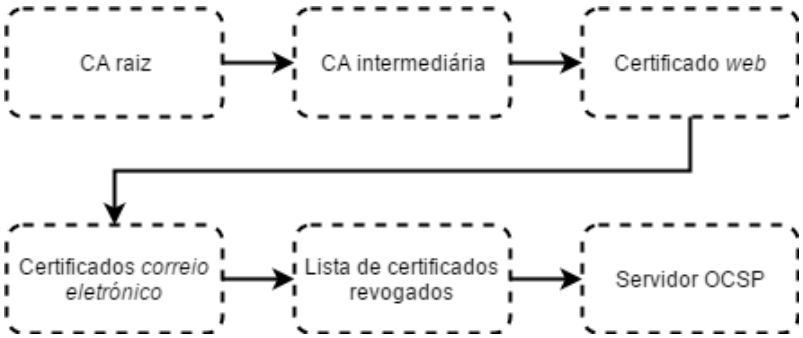


Figura 4.3: Fluxo de operações

4.3 Implementação da CA raiz

4.3.1 Geração da Chave

O primeiro passo para que uma entidade detenha um certificado que associe a sua identidade à sua chave pública, consiste na geração da chave que ficará associada ao certificado. Para este fim, foi utilizada a biblioteca *openssl* para criar um par de chaves (pública e privada) RSA de 4096 bits. A opção *-aes256* é utilizada para encriptar a chave privada utilizando encriptação simétrica AES de 256 bits. Para isso vai ser pedido ao utilizador uma *passphrase* que será a chave AES.

```
# openssl genrsa -aes256 -out ca.key.pem 4096
```

Tratando-se da uma entidade certificadora, por razões de segurança, optou-se pelo comprimento de chave de 4096 bits que providencia atualmente a maior garantia de robustez.

4.3.2 Ficheiro de Configuração

De maneira a personalizar a criação e manipulação de certificados foi necessário criar um ficheiro de configuração, `openssl_ca.cnf`, para a CA raiz. Este ficheiro é responsável pelas indicações de configurações de localizações de diretórios, políticas de emissão de certificados, configurações de solicitações e criação de certificados, e demais configurações. Os seus principais segmentos podem ser consultados no Anexo A deste documento, mais precisamente na secção A.2.

4.3.3 Geração do CSR (*Certificate Sign Request)*

Como a CA raiz é considerada a entidade confiável do sistema, não existe a necessidade de uma outra entidade assinar o seu certificado. Por essa razão esta entidade possui um certificado auto-assinado, não sendo necessária a criação de um pedido de assinatura do certificado.

4.3.4 Geração do Certificado

Com o propósito da criação e assinatura do próprio certificado para a CA raiz, foi utilizado o seguinte comando:

```
# openssl req -config openssl_ca.cnf -key ca.key.pem -new -x509 -days
7300 -extensions ca_ext -out ca.cert.pem
```

Como indicado no comando, foi utilizado o ficheiro de configuração `openssl_ca.cnf`, a chave gerada anteriormente `ca.key.pem`, o parâmetro -x509 para criar um certificado auto-assinado, o número de dias de validade (7300) e a extensão `ca_ext` definida no ficheiro de configuração. A saída do comando consiste num certificado com o nome `ca.cert.pem`.

Finalizado o processo de geração do certificado, a CA raiz encontra-se capaz de assinar certificados. No caso da estrutura implementada, a CA raiz apenas assina certificados de entidades certificadoras intermediárias, entidades essas responsáveis pela assinatura de certificados dos servidores e clientes do sistema.

4.4 Implementação da CA intermediária

4.4.1 Geração da Chave

Tal como explicitado na secção da implementação do certificado da CA raiz, de maneira a criar um certificado para a entidade certificadora emissora é necessário criar um par de chaves para essa entidade.

```
# openssl genrsa -aes256 -out sign-ca.key.pem 4096
```

4.4.2 Ficheiro de Configuração

Dada a diferença entre os processos de criação e manipulação de certificados da CA raiz e da CA intermediária, foi necessário criar um ficheiro de configuração, `openssl_sign.cnf`, para a CA intermediária. Os seus principais segmentos podem ser consultados no Anexo A deste documento, mais precisamente na secção A.2.

4.4.3 Geração do CSR

De modo a gerar um pedido de assinatura do certificado foi necessário realizar o seguinte comando:

```
# openssl req -config openssl_sign.cnf -key sign_ca.key.pem -new
-out sign_ca.csr.pem
```

Como indicado no comando foi utilizado o ficheiro de configuração `openssl_sign.cnf` e a chave gerada no passo anterior. O comando cria o ficheiro `sign_ca.csr.pem` que corresponde ao pedido de assinatura do certificado a ser enviado para a CA raiz.

4.4.4 Geração do Certificado

Após a criação do pedido de assinatura do certificado digital da CA intermediária, a requisição é enviada para a entidade responsável pela assinatura de certificados de CAs intermediárias, a CA raiz. Neste caso, o ficheiro de requisição foi criado na máquina *Debian 1* e enviado para a máquina *Debian Root*.

Na máquina *Debian Root*, foi utilizado o seguinte comando para assinar o certificado da CA intermediária:

```
# openssl ca -config openssl_ca.cnf -extensions sign_ca_ext -days 3650
-in sign_ca.csr.pem -out sign_ca.cert.pem
```

Como indicado no comando, foi utilizado o ficheiro de configuração `openssl_ca.cnf`, a extensão `sign_ca_ext` desse ficheiro, o número de dias de validade (3650) e o ficheiro `sign_ca.csr.pem` que contém o pedido da assinatura do certificado em questão. O certificado foi criado com o nome `sign_ca.cert.pem`.

4.5 Implementação do Certificado da Página *Web*

4.5.1 Geração da Chave

Geração do par de chaves para o servidor *web*, página *www.mycompany.pt*.

```
# openssl genrsa -aes256 -out www.mycompany.pt.key.pem 2048
```

4.5.2 Geração do CSR

De modo a criar um pedido de assinatura do certificado foi necessário realizar o seguinte comando:

```
# openssl req -key www.mycompany.pt.key.pem -new -out
www.mycompany.pt.csr.pem
```

Como se vê, foi utilizada a chave gerada no passo anterior. O comando cria o ficheiro `www.mycompany.pt.csr.pem` que corresponde ao pedido de assinatura do certificado a ser enviado para a CA intermediária.

4.5.3 Geração do Certificado

Após a criação do pedido de assinatura do certificado digital da página *www.mycompany.pt*, esse pedido é enviado para a entidade responsável pela assinatura de certificados de certificados para servidores, a CA intermediária. Neste caso, o ficheiro de requisição foi criado na máquina *Debian 2* e enviado para a máquina *Debian 1*.

Na máquina *Debian 1* foi utilizado o seguinte comando para assinar o certificado da página *web*:

```
# openssl ca -config openssl_sign.cnf -extensions server_cert_ext -days
375 -in www.mycompany.pt.csr.pem -out www.mycompany.pt.cert.pem
```

Como indicado no comando, foi utilizado o ficheiro de configuração `openssl_sign.cnf`, a extensão `server_cert_ext` desse ficheiro, o número de dias de validade e o ficheiro `csr` que contém a requisição da assinatura do certificado em questão. O certificado foi criado com o nome `www.mycompany.pt.cert.pem`.

O processo de instalação do servidor *web* pode ser consultado no Anexo A deste documento, mais precisamente na secção A.3.

4.6 Implementação de Certificados de Clientes

4.6.1 Geração das Chaves

Geração dos pares de chaves para dois utilizadores do sistema de correio eletrónico, *Frank* e *Sam*:

Utilizador Frank:

```
# openssl genrsa -aes256 -out frank.mycompany.key.pem 2048
```

Utilizador Sam:

```
# openssl genrsa -aes256 -out sam.mycompany.key.pem 2048
```

4.6.2 Geração dos CSRs

Requisição de assinatura de certificados para os dois utilizadores anteriores:

Utilizador Frank:

```
# openssl req -key frank.mycompany.key.pem -new -out
frank.mycompany.csr.pem
```

Utilizador Sam:

```
# openssl req -key sam.mycompany.key.pem -new -out
sam.mycompany.csr.pem
```

Como indicado nos comandos foram utilizadas as chaves geradas na etapa anterior. O comando gera o ficheiro `(nome_do_utilizador).mycompany.csr.pem` que corresponde ao pedido de assinatura do certificado a ser enviado para a CA intermediária.

4.6.3 Geração dos Certificados

Após a criação do pedido de assinatura do certificado digital dos utilizadores, os pedidos são enviados para a entidade responsável pela assinatura de certificados para utilizadores, a CA intermediária. Neste caso, os ficheiros de requisição foram criados nas máquinas *Debian 3* e *Debian 4* e enviados para a máquina *Debian 1*.

Na máquina *Debian 1*, correspondente à CA intermediária, foi utilizado o seguinte comando para assinar o certificado do utilizador *Frank*:

```
# openssl ca -config openssl_sign.cnf -extensions user_cert_ext -days 375
-in frank.mycompany.csr.pem -out frank.mycompany.cert.pem
```

Para o certificado do utilizador *Sam*:

```
# openssl ca -config openssl_sign.cnf -extensions user_cert_ext -days
375 -in sam.mycompany.csr.pem -out sam.mycompany.cert.pem
```

Como indicado nos comandos, foi utilizado os ficheiro de configuração `openssl_sign.cnf`, a extensão `user_cert_ext` desse ficheiro, o número de dias de validade (375) e o ficheiro `csr` que contém a requisição da assinatura do certificado em questão. Os certificados foram criados com os nomes `frank.mycompany.cert.pem` e `sam.mycompany.cert.pem`.

O processo de instalação do servidor de correio eletrónico pode ser consultado no Anexo A deste documento, mais precisamente na secção A.3.

4.7 Implementação da Lista de Certificados Revogados

4.7.1 Revogação de um Certificado

No processo de assinatura ou revogação de um determinado certificado o sistema *openSSL* atualiza o ficheiro `index.txt`. Este ficheiro funciona como uma base de dados que contém informação sobre todos os certificados administrados pela CA.

Quando a CA assina um certificado, é adicionado um registo ao ficheiro com as informações principais do certificado. Quando a CA revoga um certificado, o estado desse certificado, no ficheiro, é alterado para revogado.

A título de exemplo, foi revogado um certificado utilizado para testes, `www.test-mycompany.pt.cert.pem`, previamente assinado pela CA intermediária.

De modo a revogar o certificado, foi necessário realizar o seguinte comando:

```
# openssl ca -config openssl_sign.cnf -revoke
www.test-mycompany.pt.cert.pem
```

O comando faz com que o registo referente ao certificado *www.test-mycompany.pt* do ficheiro `index.txt` se altere, indicando a invalidade do mesmo.

De seguida apresenta-se um excerto da informação contida no ficheiro `index.txt`.

```
V 180316045555Z 1007 (...)/ CN=admin@mycompany.pt/(...)
V 180316045622Z 1008 (...)/CN=frank@mycompany.pt/(...)
R 180316045715Z 170317030446Z 100A (...)/CN=www.test-mycompany.pt/(...)
```

A informação anterior constata que a linha referente ao certificado *www.test-mycompany.pt* possui o indicador de estado a "R", significando que se trata de um certificado revogado. Todos os outros certificados têm o indicador a "V", tratando-se de certificados válidos. Os campos a sublinhado correspondem aos números de série de cada registo.

4.7.2 Geração da Lista de Certificados Revogados

De modo a gerar a lista de certificados revogados, foi utilizado o seguinte comando na máquina que aloja a CA intermediária, *Debian 1*:

```
# openssl ca -config openssl_sign.cnf -gencrl -out sign-ca.crl.pem
```

O comando gera o ficheiro *sign-ca.crl.pem* que contém informações sobre o emissor do ficheiro, período de validade do ficheiro e todos os certificados revogados pela CA. Os certificados são identificados pelo seu número de série.

O processo de instalação da lista de certificados revogados pode ser consultada no Anexo A deste documento, mais precisamente na secção A.3.

4.8 Implementação do Certificado do Servidor OCSP

4.8.1 Geração da Chave

O facto de que as respostas realizadas pelo servidor OCSP tenham que ser assinadas digitalmente, origina que a própria entidade necessite de um par de chaves de modo a possuir um certificado digital.

O seguinte comando foi utilizado para gerar o par de chaves correspondente:

```
# openssl genrsa -aes256 -out ocsp.mycompany.key.pem 4096
```

4.8.2 Geração do CSR

De modo a criar um pedido de assinatura de certificado foi necessário realizar o seguinte comando:

```
# openssl req -key ocsp.mycompany.key.pem -new -out
ocsp.mycompany.csr.pem
```

Como indicado no comando foi utilizada a chave criada no passo anterior. O comando gera o ficheiro `ocsp.mycompany.csr.pem` que corresponde ao pedido de assinatura do certificado a ser enviado para a CA intermediária.

4.8.3 Geração do Certificado

Após a criação do pedido de assinatura do certificado digital do servidor OCSP, esse pedido é enviado para a entidade responsável pela assinatura de certificados de servidores, a CA intermediária. Neste caso, o ficheiro de requisição foi enviado para a máquina *Debian 1*. Nessa máquina foi utilizado o seguinte comando para assinar o certificado do servidor OCSP:

```
# openssl ca -config openssl_sign.cnf -extensions ocsp_ext -days 375 -in
ocsp.mycompany.csr.pem -out ocsp.mycompany.cert.pem
```

Como indicado no comando, foi utilizado o ficheiro de configuração `openssl_sign.cnf`, a extensão `ocsp_ext` desse ficheiro, o número de dias de validade e o ficheiro `csr` que contém a requisição da assinatura do certificado em questão. O certificado foi criado com o nome `ocsp.mycompany.cert.pem`.

O processo de instalação do servidor OCSP pode ser consultado no Anexo A deste documento, mais precisamente na secção A.3.

4.9 Considerações

Através do processo de implementação apresentado foi possível constatar, de uma forma prática, a possibilidade da criação de uma estrutura de distribuição de chaves públicas através de certificados digitais evitando a utilização de CAs externas. O processo requer um certo grau de conhecimento no que diz respeito às técnicas utilizadas mas permite evidenciar que existem formas que permitem eliminar a necessidade de entidades externas para certificações de relações entre entidades e as suas chaves criptográficas públicas. Considera-se que a implantação de um serviço como o descrito enquadra-se dentro do grau de conhecimentos da equipa informática da U.Porto e o número de utilizadores justificavam tal serviço.

Capítulo 5

Análise Universidade do Porto

Este capítulo tem como objetivo realizar uma análise do sistema de certificação existente na U.Porto, bem como a apresentação de alterações e considerações que têm o intuito de melhorar o supracitado sistema. Primeiramente é exposto um estudo sobre o estado da certificação digital da U.Porto, seguido de um conjunto de sugestões que têm como base a utilização de uma infraestrutura interna de distribuição de chaves públicas na U.Porto. Por último procede-se a um estudo sobre a possibilidade da não utilização de certificados digitais em sistemas de distribuição de chaves públicas.

5.1 Estado Certificação Digital U.Porto

A Universidade do Porto é uma instituição pública de ensino e investigação situada na cidade do Porto, Portugal. Fundada em 1911, é composta por 14 faculdades, uma *business school* e mais de 50 centros de investigação[30]. Atualmente a Universidade não dispõe de um serviço PKI próprio, ao invés, utiliza os serviços de organizações externas, como a *DigiCert* e a *Comodo*, para assinar certificados digitais dos seus próprios serviços e utilizadores internos.

No presente, a U.Porto disponibiliza o acesso a certificados digitais emitidos, por parte da *DigiCert*, ao abrigo do projeto TCS (*Trusted Certificate Service*). O TCS consiste num serviço que tira partido de aquisições em massa de certificados digitais por parte de instituições de educação e investigação. O serviço disponibiliza os seguintes tipos de certificados digitais[31]:

- Certificados Pessoais *Grid*: Certificados digitais pessoais *eScience* que podem ser usados em serviços *Grid*;
- Certificados Pessoais: Certificados para assinatura digital e certificados para segurança de correio eletrónico;
- Certificados de Servidor: Certificados para máquinas servidoras.

Na figura 5.1 encontra-se representado o certificado digital correspondente à entidade *sigarra.up.pt*, que consiste num serviço *web* disponibilizado pela U.Porto para o acesso a todo o sistema de informação da universidade, assinado pela entidade externa *DigiCert*.

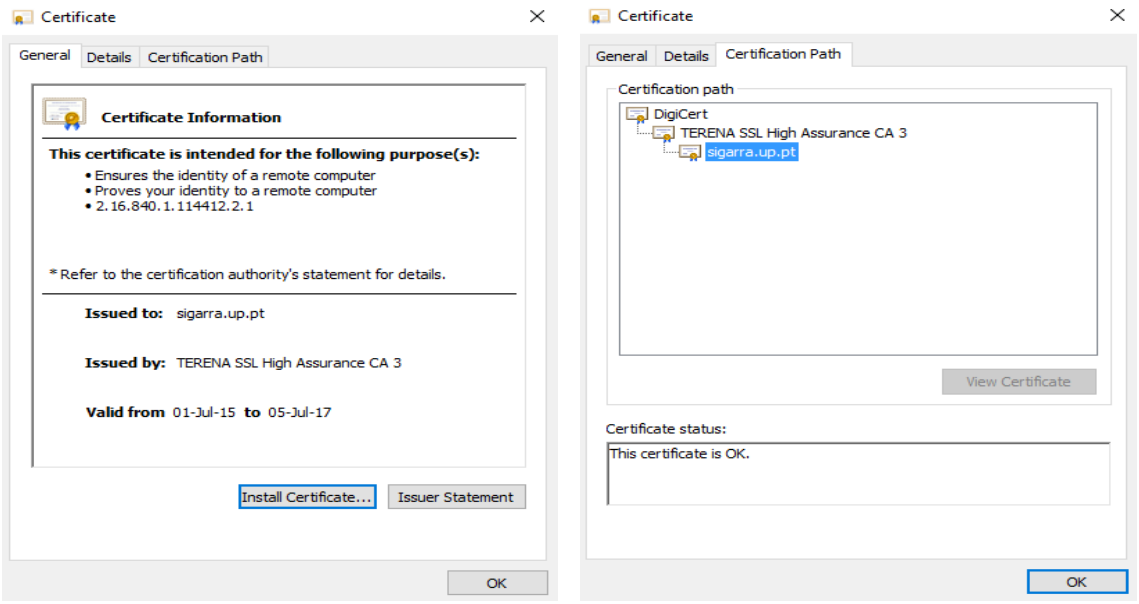


Figura 5.1: Certificado digital *sigarra.up.pt*

Na figura 5.2 encontra-se representado o certificado digital correspondente à entidade *diplomas.up.pt*, que consiste num serviço de documentos institucionais eletrónicos *online* disponibilizado pela U.Porto, assinado pela entidade externa *UserTrust*, pertencente ao grupo *Comodo*.

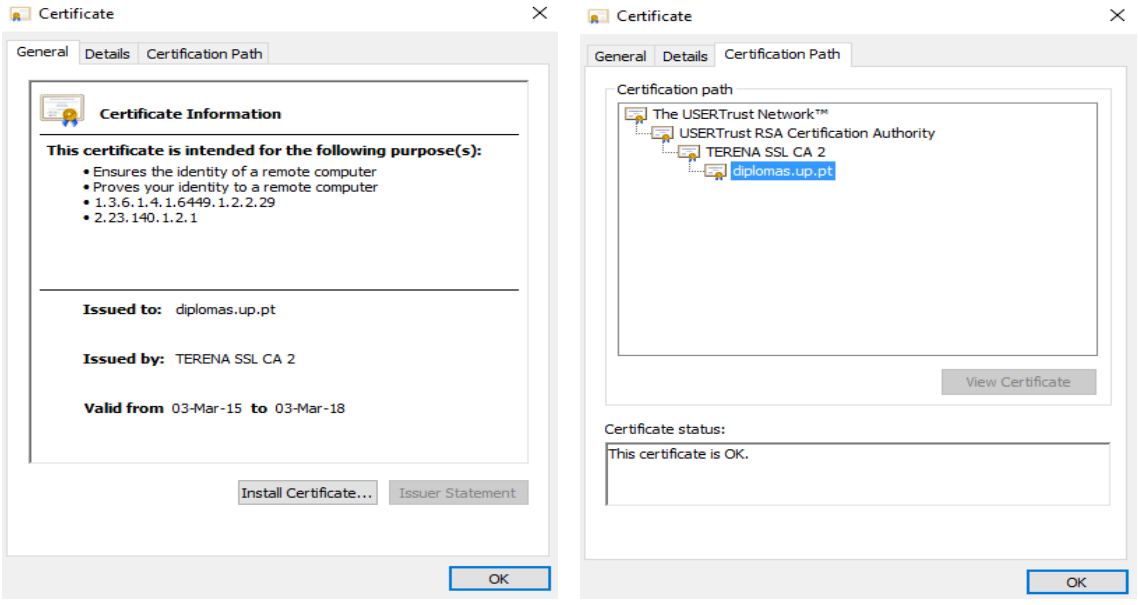


Figura 5.2: Certificado digital *diplomas.up.pt*

5.1.1 Pedido de um Certificado Digital Pessoal

Para efeitos de requisição de um certificado digital pessoal disponibilizado pela U.Porto, acedeu-se à página UPdigital[1]. Na secção reservada aos certificados digitais procedeu-se à solicitação do certificado através de uma hiperligação presente na página que indica a utilização do portal da *Digicert* para a solicitação do serviço.

Para continuar o processo, foi necessário requisitar um acesso autorizado ao portal da *Digicert* através de um correio eletrónico enviado para o endereço *csirt@uporto.pt*. Dois dias volvidos, foi recebido o respetivo acesso com as credenciais através de um correio eletrónico enviado pelos serviços de segurança da *Comodo*. O passo seguinte consistiu na requisição do certificado através de uma hiperligação indicada no correio eletrónico. Segundos após o término do processo de aplicação, o certificado foi instalado automaticamente no navegador *web*.

A figura 5.3 representa o certificado entregue pelo serviço para o utilizador *ee11293@fe.up.pt*.

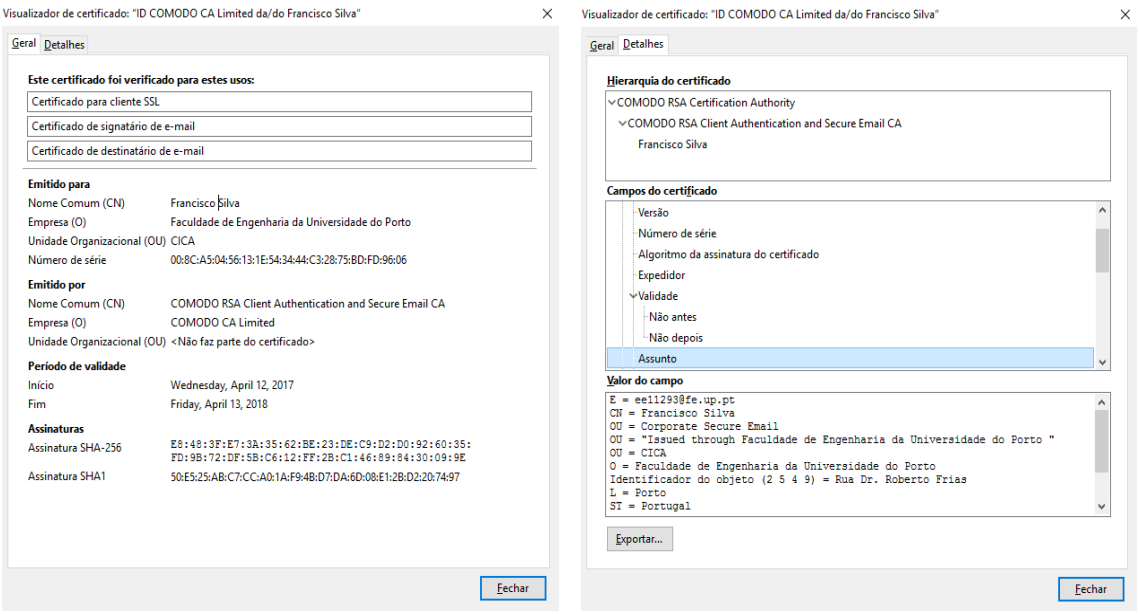


Figura 5.3: Certificado digital: ee11293@fe.up.pt

5.2 Sugestões de Certificação

Tendo a U.Porto uma grande unidade de serviços de Informática (UPdigital), julga-se que existem condições técnicas para poder efetuar serviços de certificação interna. A seguir apresentam-se algumas sugestões de *software* que podem ser utilizados para este fim. Depois, realizam-se uma série de observações que apoiam a validade da ideia da utilização de uma PKI interna na U.Porto.

[1] **https://sigarra.up.pt/reitoria/pt/web_base.gera_pagina?P_pagina=1018486**

5.2.1 Implementação de uma Estrutura Interna de Certificados

Como comprovado no capítulo 4 do presente documento, é possível conceber um sistema de criação e gestão de certificados digitais através da biblioteca *openSSL*. Esta ferramenta, *open source*, permite a administração de uma estrutura de certificados através de linha de comandos e oferece os serviços necessários à sua gestão, tais como a geração de chaves, criação de pedidos de assinatura de certificados, assinatura de certificados, revogação de certificados, diferenciação de tipos de certificados e ações permitidas, etc.

Apesar do alargado conjunto de funcionalidades oferecidas pelo *openSSL*, a sua implementação consiste num exercício de baixo nível, sendo necessária a manipulação direta da interface de linha de comandos.

Constituindo uma implementação de dimensão reduzida ficou comprovado que, para uma estrutura de dimensão alargada, a aplicação crua da biblioteca *openSSL* pode causar inúmeras dificuldades, devido a questões de escalabilidade e interface. De modo a solucionar o problema de baixo nível das implementações diretas em *openSSL*, existem disponíveis várias soluções que implementam operações do tipo PKI. De seguida, é apresentada uma análise superficial de algumas soluções existentes.

5.2.1.1 OpenCA

O projeto OpenCA[2] teve início em 1999 consistindo numa aplicação gratuita para a implementação de uma infraestrutura de chaves públicas conseguida através de uma interface *web* Perl e uma base de dados, sendo as operações criptográficas realizadas através da biblioteca *openSSL*.

A solução baseia-se em vários projetos *open source*, entre os quais *openLDAP*, *openSSL*, *Apache*, e encontra-se atualmente disponível para os sistemas operativos *Linux*, *Solaris* e *MacOs X*. A última atualização ao *sofware* disponibilizado para descarregamento foi em agosto de 2013[32].

Principais serviços disponibilizados:

- Interface CA;
- Interface RA (Autoridade de Registo);
- LDAP (*Lightweight Directory Access Protocol*);
- OCSP;
- Emissão de listas de certificados revogados;
- *Login* baseado em certificados;
- Integração com HSM (*Hardware Security Module*);
- Suporte a múltiplas bases de dados.

[2] **https://www.openca.org/projects/openca/**

5.2.1.2 EJBCA

O EJBCA[3] (*Enterprise Java Beans Certificate Authority*) trata-se de uma solução gratuita que visa a implementação de uma CA numa estrutura de certificados. Este *software*, construído através de tecnologia JAVA consiste numa implementação escalável, robusta e flexível, capaz de realizar a gestão de uma infraestrutura de chaves públicas de dimensões consideráveis com múltiplos níveis de hierarquias. Para além dos serviços normais de uma PKI, destacam-se a independência às plataformas do sistema hospedeiro, o suporte a diferentes arquiteturas (unificada, *clustered*, RA externa, OSCP externo, etc), suporte a múltiplas bases de dados, entre outras.

Como referências de sucesso de instalações do EJBCA evidenciam-se o ministério da defesa de França, ministério das finanças de França, polícia nacional da Suécia e a Universidade de Ciências aplicadas de Zurique[33].

Principais serviços disponibilizados:

- Múltiplas CAs e vários níveis de hierarquia;
- Administração através de *web* GUI, linha de comandos ou serviços *web*;
- Múltiplos níveis de administração;
- OCSP;
- LDAP;
- Emissão de listas de certificados revogados;
- Gestão de perfis (tipos) de certificados;
- Integração com HSM;
- Implementação de certificados *smartcard logon*.

5.2.1.3 Active Directory Certificate Services (AD CS)

A solução AD CS[4] consiste num *software* pertencente à *Microsoft* para a plataforma *Windows*, que permite a implementação de uma infraestrutura de chaves públicas governável, escalável e segura, disponibilizado a partir do *Windows Server 2000*.

O AD CS suporta aplicações do tipo S/MIME, VPN, IPSec, TLS, assinaturas digitais, *smart card logon*, etc. Constituindo um serviço *Windows* necessita de uma licença (*Windows Server*) para poder ser utilizado[34].

[3] **https://www.openca.org/projects/openca/**
[4] **https://technet.microsoft.com/en-us/library/cc732625(v=ws.11).aspx**

Principais serviços disponibilizados:

- Interface Pública *web*;
- Múltiplas CAs e vários níveis de hierarquia;
- Autoridades de registo;
- OCSP;
- LDAP;
- Emissão de listas de certificados revogados;
- Serviços de registo de componentes de rede.

5.2.2 Justificação da Viabilidade da Manutenção de uma PKI Interna

Perante a sugestão da utilização de uma estrutura de certificados interna, são apresentados alguns aspetos que permitem justificar a sua viabilidade.

Independência de CAs externas

A capacidade de independência a CAs externas, para além de evitar todos os possíveis problemas causados pelas suas vulnerabilidades, permite criar um sistema privado totalmente controlado pela entidade reguladora.

Acordos com Instituições

Dado o facto da U.Porto ser uma instituição de ensino, a possibilidade de uma expansão da estrutura através de acordos com outras instituições provocaria um aumento na segurança das comunicações entre utilizadores e entre os próprios serviços das instituições incluídas nos acordos.

Estes acordos teriam especial interesse com instituições que também detivessem a sua própria estrutura de certificados e seriam possíveis através da implementação de "certificados cruzados". Este tipo de certificados permitem que entidades de uma determinada estrutura passem a confiar em certificados de uma outra estrutura de certificados.

Repositório Público

Uma das maiores restrições à utilização de certificados digitais para prover segurança às comunicações entre entidades consiste na falta de meios para obter certificados digitais das entidades com as quais se pretende comunicar. Num sistema de proporções gigantes como a Internet, a possibilidade da existência de um repositório público central com os certificados de todas as entidades é uma tarefa difícil de efetivar devido à dimensão do repositório e do esforço dos processos de validação dos certificados.

Relativamente a um sistema fechado como o da U.Porto, a utilização de um repositório público de certificados digitais faria com que os próprios utilizadores dos serviços do sistema detivessem a possibilidade de obtenção do certificado digital referente à entidade (servidores, utilizadores finais, etc) com a qual pretendem comunicar dentro do sistema da U.Porto. A sua implementação poderia ser realizada através de um sítio *web* especializado para a obtenção dos certificados, da página pessoal de cada utilizador (no caso de certificados de utilizadores), através da tecnologia LDAP, etc.

5.3 Análise Estrutura sem Certificados

De seguida é exposto um estudo sobre a aplicação de uma estrutura de chaves públicas sem a necessidade da utilização de certificados digitais. Nas PKI tradicionais, os certificados digitais atuam como um meio para atingir uma comprovação em como a chave pública pertence de facto a uma determinada entidade e onde as diversas aplicações utilizam as chaves públicas extraídas dos certificados digitais para providenciar segurança às suas comunicações através de técnicas de criptografia assimétrica. A possibilidade de ignorar os serviços fornecidos pelos certificados digitais através de um outro método que estabeleça a associação entre uma entidade e a sua chave pública permitiria a libertação da necessidade das atuais entidades certificadoras.

Seguidamente apresentam-se três possíveis métodos, juntamente com uma breve análise, que constituem uma forma de evitar a utilização de certificados digitais. Para todos os casos, existiria a necessidade de alterar a comunicação TLS de modo a que o servidor envie a sua chave pública ao cliente e não o seu certificado digital.

5.3.1 Base de Dados Local

O protocolo SSH implementa uma alternativa à utilização de certificados que consiste na existência em disco, do lado dos utilizadores, de uma base de dados com associações de entidades às suas correspondentes chaves públicas[35].

PKI Interna

Este tipo de solução, é plausível para sistemas onde os utilizadores apenas necessitam de comunicar com um número limitado de entidades, visto que para qualquer comunicação com uma nova entidade, é necessário confiar cegamente na chave pública ou derivá-la de uma outra forma (presencial, correio eletrónico, etc).

PKI Global

Para sistemas em que existe um número ilimitado de comunicações com outras entidades, como a Internet, a solução torna-se incomportável, podendo comprometer a segurança das comunicações devido à maior probabilidade de ocorrências de ataques MITM.

5.3.2 Repositórios Públicos

A utilização de repositórios públicos que armazenem informação de associações de entidades à sua chave pública consiste num outro tipo de solução para a não utilização de certificados digitais. Este tipo de repositório, de acesso seguro, permitira ao utilizadores obter a chave pública de uma entidade através de uma consulta aos seus registos. Tomando como premissa a fiabilidade na entidade responsável pelo repositório, o que faz todo o sentido em redes locais como a da FEUP ou da U.Porto, a sua utilização permitiria descartar a utilização de certificados digitais, através de um serviço simples e direto.

PKI Interna

Este tipo de solução, para estruturas fechadas de chaves públicas, consiste numa proposta plausível para abdicar dos serviços dos certificados digitais. A dimensão limitada de registos e o limitado número de acessos simultâneos permitem uma obtenção eficiente da informação. O número limitado de entidades permite a unicidade dos registos de entidades no repositório, tornando o processo de registo e manutenção possível.

PKI Global

Para sistemas em que o número de registos nos repositórios fosse ilimitado, como o caso da Internet, um repositório global consistiria num serviço pouco eficiente em termos de desempenho e manutenção. O serviço resultaria numa base de dados com inúmeros registos, alvo de inúmeros acessos simultâneos tornando a sua usabilidade difícil. Uma outra razão consiste na unicidade dos identificadores das entidades. O número ilimitado de entidades num repositório dificultaria o processo de registo de entidades na base de dados devido à complexidade na obtenção de um identificador único, concreto e legível que permitisse reconhecer inequivocamente uma entidade. Este processo é crucial para não existir num mesmo repositório referências a entidades diferentes com o mesmo identificador.

5.3.3 DANE com Chave Pública

O protocolo DANE, brevemente explicitado no capítulo 2 e documentado com mais detalhe no capítulo 6 deste documento, permite a indicação da chave pública do servidor em vez do seu certificado digital no registo DNS do domínio em questão[8]. Este procedimento possibilitaria aos detentores dos domínios a indicação da chave pública a comparar com a chave pública entregue pelo servidor no processo de comunicação cliente-servidor. A utilização deste aspeto providenciaria ao sistema independência das CAs e eliminação da necessidade de certificados digitais.

Quanto ao seu funcionamento, na consulta DNS, o cliente para além do endereço de IP também receberia a informação da chave pública do servidor com o qual pretende comunicar. No processo de comunicação com o servidor, no momento da receção da chave pública (no processo normal seria um certificado digital), o cliente realizaria a comparação dessa chave com a chave obtida na consulta DNS de modo a validar o acesso ao servidor. Se as chaves forem idênticas o acesso é

considerado válido; se não existir correspondência, então a chave pública recebida do servidor não é a indicada pelo detentor do domínio e o acesso deverá ser considerado inseguro. Esta solução requisitaria a implementação de métodos de validação das tecnologias DNSSEC e DANE por parte de todos utilizadores que pretendam aceder aos servidores que a implementam.

5.4 Considerações

Tendo em conta o número de utilizadores e os serviços disponibilizados pela U.Porto, encara-se com apreensão o fato desta instituição depender de organizações externas para providenciar comunicações internas seguras aos seus utilizadores. A possibilidade de uma estrutura de certificação interna controlada pela própria U.Porto permitiria, entre muitas conveniências, a abolição da necessidade de contratos com autoridades certificadoras para a emissão de certificados digitais que são primariamente usados dentro na Universidade pelos seus membros no acesso aos seus serviços. Uma vez que o princípio essencial de qualquer serviço de certificados se baseia na confiança existente na entidade raiz, a imposição de uma relação de confiança com uma entidade desconhecida (*Comodo*, *Digicert*, etc.) pode ser classificada como pouco compreensível quando, por outro lado, existe uma relação de confiança pré-estabelecida com a própria U.Porto.

Quanto ao estudo de uma estrutura sem certificados, considera-se que a utilização da solução DANE consistiria na opção com maior validade devido à utilização da comprovada tecnologia DNS com DNSSEC e à promoção de uma fácil adaptação ao sistema atual possibilitando uma alteração de paradigma em fases. Primeiramente com a implementação da solução DANE com certificados digitais e numa fase posterior a continuação da sua implementação sem a aplicação de certificados digitais.

Capítulo 6

Análise Objetiva Otimizações/Alternativas

Neste capítulo realiza-se uma análise detalhada de três otimizações/alternativas ao sistema atual de distribuição de chaves públicas. Para cada otimização/alternativa é efetuada uma exposição geral, uma descrição do seu funcionamento, análise de pontos fortes e pontos fracos, e uma indicação sobre o seu estado atual de utilização geral e do estado de implementação na U.Porto.

Este capítulo expande o que já foi sucintamente apresentado no capítulo 2, nomeadamente na secção 2.3.

6.1 DANE

O DANE consiste num protocolo que visa vincular os certificados digitais utilizados na infraestrutura de chaves públicas ao sistema DNS utilizando o serviço DNSSEC de modo a providenciar autenticação às mensagens enviadas pelos servidores do sistema DNS.

Numa perspetiva superficial, o protocolo propõe que as entidades detentoras dos domínios associem, através de DNSSEC, um certificado a um registo DNS de modo a adicionar uma camada de segurança ao sistema.

Os conceitos base desta solução encontram-se explicitados na subsecção 2.3.1 deste documento.

6.1.1 Funcionamento

Como mencionado anteriormente, o protocolo DANE faz uso do serviço e estrutura do DNS e da capacidade do DNSSEC em providenciar segurança. A tecnologia DANE oferece um método de autenticar a associação do certificado de um servidor com o seu domínio através da inclusão de uma indicação de um certificado num registo DNS. Desta forma, a associação entre um servidor e a sua chave pública é providenciada ao cliente através do serviços DNS protegido por DNSSEC.

Relativamente ao protocolo TLS, a especificação DANE cria um novo tipo de registo (TLSA) que consiste num elemento que associa um certificado ou a chave pública de um servidor com o

nome do domínio onde o registo está inserido, indicando o certificado que deverá ser utilizado pelo cliente para validar o acesso ao servidor. O formato do registo é estruturado em quatro campos: aplicação do certificado, seletor, tipo de correspondência e o campo referente a dados de associação do certificado[8].

Campo "Aplicação do certificado"

Indicação do método a utilizar para validar o certificado providenciado nas primeiras trocas de informação TLS, utilizando a informação contida no campo de associação do certificado.

0 – PKIX-TA — Especifica um certificado de uma CA (ou a chave pública desse certificado) que deverá ser encontrado no caminho de certificados do certificado entregue pelo servidor na comunicação TLS. Para além desta validação, o certificado também deverá ser verificado através da validação tradicional do caminho de certificados (PKI).

1 – PKIX-EE — Especifica um certificado de uma entidade final (ou a chave pública desse certificado) que deverá corresponder ao certificado entregue pelo servidor na comunicação TLS. Para além desta validação, o certificado também deverá ser verificado através da validação tradicional do caminho de certificados.

2 – DANE-TA — Especifica um certificado (ou a chave pública desse certificado) que deverá ser utilizado como a referência de confiança na validação do certificado entregue pelo servidor na comunicação TLS. Este método é útil se um determinado domínio emitir os seus próprios certificados utilizando a sua própria CA (que não estará intrinsecamente referenciada como uma entidade de confiança para os utilizadores);

3 – DANE-EE — Especifica um certificado (ou a chave pública desse certificado) que deverá corresponder ao certificado da entidade final entregue pelo servidor na comunicação TLS. Este método difere do (1) na medida em que não é necessária a validação do caminho de certificado PKI, permitindo a utilização de certificados auto-assinados.

Campo "Seletor"

Especifica que parte do certificado entregue pelo servidor na comunicação TLS é comparado com os dados de associação do certificado presente no registo TLSA.

0 – Certificado completo;
1 – Campo *SubjectPublicKeyInfo* do certificado;

Campo "Tipo de correspondência"

Especifica como a avaliação da correspondência entre a informação de associação do certificado e o certificado entregue pelo servidor na comunicação TLS é realizada.

0 – Correspondência exata
1 – *Hash* SHA-256
2 – *Hash* SHA-512

Campo "Dados de associação do certificado"

Especifica os dados do registo DNS a serem comparados com o certificado entregue pelo servidor na comunicação TLS.

De seguida apresenta-se um exemplo de um registo TLSA correspondente a um servidor HTTP sobre TLS na porta 443, com a indicação da aceitação somente de certificados emitidos pela CA especificada com verificação PKI, certificado completo para comparação, com correspondência através de uma *hash* SHA-256 e indicação da *hash* do certificado.

```
_443._tcp.www.example.com.  IN
TLSA (0 0 1 67877f4ec3a2a1b57aa2e91d74e48cd40e616d401ec276048c089afedd64
550b)
```

6.1.2 SMTP

A solução DANE pode também ser utilizada no protocolo SMTP de modo a melhorar os aspetos relativos à segurança das comunicações em si suportadas.

Um dos maiores problemas de segurança do sistema SMTP consiste na possibilidade de ataques de *downgrade*. No protocolo HTTP a utilização da sua versão segura (TLS) é indicada através do esquema URI, ao contrário do protocolo SMTP que não detém uma forma de indicação para a utilização TLS. Como solução, o protocolo SMTP recorre a um modelo oportunístico para o modelo de segurança, onde ambas as partes da comunicação negoceiam entre si a disponibilidade para uma comunicação TLS. O problema desta abordagem consiste na possibilidade que um utilizador, por exemplo um atacante MITM, detém em "recusar" a indicação do servidor em utilizar o protocolo TLS e assim iniciar uma comunicação insegura [36].

Através do DANE e da consequente presença de um registo TLSA nos registos DNS, os servidores SMTP podem confirmar a disponibilidade de comunicações TLS, prevenindo ataques de *downgrade*.

Quanto ao seu funcionamento, as operações são semelhantes ao processo HTTPS. Tome-se como exemplo um utilizador A que envia um correio eletrónico para um utilizador B. Para o envio da mensagem, o *software* de correio eletrónico de A envia os dados necessários para o seu MTA de modo a ser esta entidade a tratar do processo de envio. Seguidamente o MTA do utilizador A requisita o registo TLSA do MTA destinatário ao seu serviço DNS de modo a validar o seu certificado, e assim iniciar uma comunicação segura. Desta forma, se um registo TLSA for encontrado pelo MTA para o servidor de destino, então a comunicação terá que ser obrigatoriamente através do protocolo TLS.

A respeito do campo de "Aplicação do certificado", a implementação SMTP apresenta uma variante que consiste na recomendação da não utilização dos valores 0 e 1.

A falta de capacidade dos MTAs em realizar verificações de identidade ou prevenir ataques de *downgrade* sem depender do DNSSEC faz com que a utilização de verificações de confiança através do esquema PKI de CAs sejam inúteis. As principais causas centram-se na possibilidade que um atacante, capaz de comprometer a implementação DNSSEC, tem em alterar os registos TLSA correspondentes ao valor de "Aplicação do certificado" para valores do seu interesse, e pelo fato de existir uma limitação do sistema associada à impossibilidade de uma interação com o cliente para resolver o problema referente a CAs que não são consideradas confiáveis [36].

De seguida apresenta-se um exemplo de um registo TLSA correspondente a um servidor SMTP sobre TLS na porta 25, com a indicação da aceitação somente de certificado presente no registo TLSA, *SubjectPublicKeyInfo* do certificado para comparação, com correspondência através de uma *hash* SHA-256 e indicação da *hash* do certificado.

```
_25._tcp.mail.example.com.  IN
TLSA (3 1 1 d8b5a11a69f68cf599e3bc53632991471a4036e09281ed8f87074f94990
cdee2)
```

6.1.3 Pontos Fortes

De seguida apresentam-se os aspetos mais interessantes da especificação DANE no que diz respeito ao foco deste documento:

- Controlo dos certificados

  Uma das grandes falhas do sistema atual consiste na possibilidade da criação, por parte de qualquer CA, de um certificado digital para um determinado domínio sem a sua autorização. Este problema faz com que os utilizadores possam tomar como válidos certificados assinados por entidades certificadoras nas quais confiam mas que não foram autorizadas pelos detentores do domínio para a emissão dos certificados. O problema torna-se mais grave quando o sistema de uma CA é comprometido de tal forma que permita um atacante a emissão de certificados em seu nome.

  O DANE permite que os detentores de domínio indiquem o método pelo qual deve ser validado o certificado proveniente do domínio em comparação com o certificado indicado no registo TLSA. Este processo confere ao detentor do domínio um controlo sobre os certificados tomados como válidos para o acesso ao seu domínio, solucionando o problema de emissão indiscriminada de certificados por parte das CAs.

- Certificados auto-assinados

  A possibilidade de ultrapassar a necessidade de utilização de CAs para a emissão de certificados através da associação de certificados auto-assinados consiste numa das grandes vantagens do protocolo DANE. Neste conceito, a confiança dos utilizadores no sistema DNS

e a utilização de DNSSEC permite garantir que a informação do registo TLSA se trate de informação legítima e autêntica.

No protocolo, os detentores dos domínios têm a possibilidade de indicar o certificado digital auto-assinado no registo TLSA do seu domínio, proporcionando aos seus clientes uma possível validação do certificado descarregado no servidor com o do registo TLSA sem a utilização de entidades externas.

- Capacidade de utilização de chaves públicas

  No RFC6698, a utilidade do registo TLSA é definida como:

  "O registo DNS TLSA é utilizado para associar um certificado TLS de um servidor ou chave pública com o nome de domínio onde o registo é encontrado, formando uma associação de certificado TLSA" [8].

  Como indicado na citação, o protocolo permite a indicação de uma chave pública em vez de um certificado digital no registo TLSA. Este aspeto pode ser interessante na possível exploração de métodos alternativos aos certificados digitais, na medida em que oferece uma forma de evitar a sua utilização através da indicação direta da chave pública a ser utilizada pelos clientes.

- Resolução do problema de *downgrade* em SMTP

  Como explicitado anteriormente, a existência nos registos DNS do registo TLSA referente ao servidor SMTP impossibilita a ocorrências de ataques onde a comunicação TLS é recusada de uma forma propositada de modo a estabelecer comunicações inseguras.

6.1.4 Pontos Fracos

Por outro lado, apresentam-se os aspetos mais inconvenientes de uma aposta na solução DANE:

- Dependência do estado de implementação do DNSSEC

  Uns dos maiores entraves ao crescimento em termos de utilizadores do protocolo DANE consiste na sua dependência ao estado de implantação do DNSSEC. Apesar da utilização massiva do sistema DNS, a implementação do DNSSEC ainda não é adotada por uma grande quantidade de entidades, sendo a sua complexa instalação vista como um dos maiores obstáculos à utilização em massa do DNSSEC[37].

- Aumento de latência nos acessos

  A utilização dos sistemas de DNSSEC para a autenticação das mensagens DNS e DANE para a validação dos certificados digitais tem efeito nos índices de desempenho devido às verificações extras, em comparação com o sistema atual.

- Erros nas configurações

  A configuração, por parte dos detentores dos domínios, dos registos referentes ao DNSSEC e DANE constitui um problema devido à possibilidade de erros nas configurações. Como, neste sistema, os acessos são validados através da comparação da informação presente no certificado e a informação presente no registo DNS, qualquer falha na configuração do sistema DNS resultará na indisponibilidade do serviço.

- Centralização do sistema

  Uma das maiores desvantagens do protocolo DANE é a permanência da existência de um sistema centralizado onde é imposta uma relação de confiança para com os servidores raiz de modo a ser possível considerar autênticas as comunicações DNS. Tal como no sistema atual PKI, onde todo o sistema depende cegamente nas ações de uma quantidade limitada de organizações (CAs), a implementação DANE depende de 13 organizações que atualmente gerem os servidores da zona raiz do sistema DNS.

  A lista atual de servidores raiz proporcionada pela Internet Assigned Numbers Authority[38] está representada na tabela 6.1.

Tabela 6.1: Lista de servidores raiz

Servidor	**Administrador**
a.root-servers.net	VeriSign, Inc.
b.root-servers.net	University of Southern California (ISI)
c.root-servers.net	Cogent Communications
d.root-servers.net	University of Maryland
e.root-servers.net	NASA (Ames Research Center)
f.root-servers.net	Internet Systems Consortium, Inc.
g.root-servers.net	US Department of Defense (NIC)
h.root-servers.net	US Army (Research Lab)
i.root-servers.net	Netnod
j.root-servers.net	VeriSign, Inc.
k.root-servers.net	RIPE NCC
l.root-servers.net	ICANN
m.root-servers.net	WIDE Project

6.1.5 Estado de Utilização

Devido à dependência da tecnologia DNSSEC por parte do DANE, primeiramente é exposta uma análise ao seu estado de implementação.

Existem dois pontos principais para a análise da implementação do DNSSEC que consistem na validação DNSSEC e na assinatura DNSSEC.

Estado da Validação DNSSEC

A validação DNSSEC consiste na utilização dos serviços de DNSSEC depois de estes serem implementados com sucesso.

A figura 6.1, disponibilizada pela página *APNIC Labs*, representa a taxa de validação mundial de DNSSEC em dezembro de 2016 para a validação do sistema DNS, onde se destacam a Suécia com 79% e a China com 2%. Como é possível verificar na figura, a distribuição da taxa de utilização não é homogénea, sendo possível identificar áreas de contraste elevado de valores de validação DNSSEC.

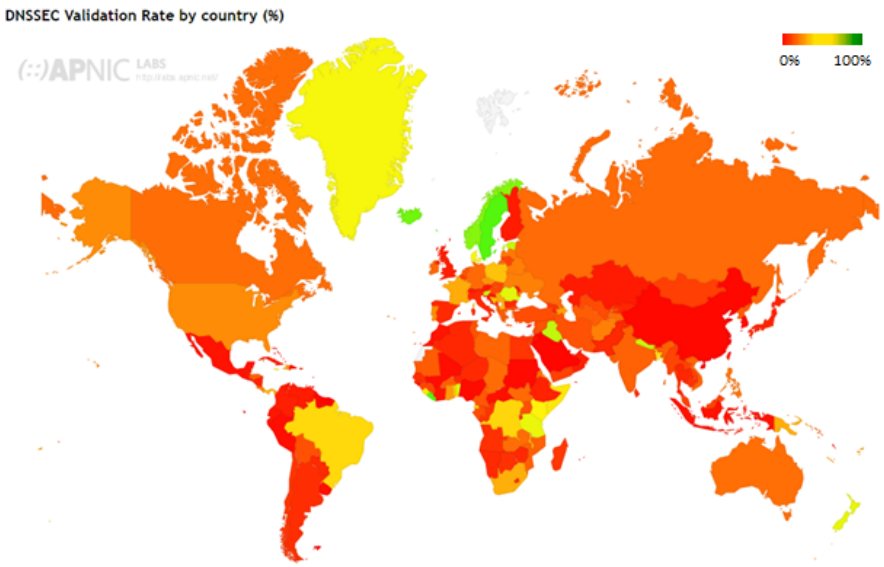


Figura 6.1: Taxa de validação mundial de DNSSEC (12/2016)[39]

A figura 6.2, também disponibilizada pela página *APNIC Labs*, evidencia o crescimento global da validação entre 2014 e inícios de 2017. É possível verificar um crescimento notório na média global de validação DNSSEC, de menos de 10% em janeiro de 2014 para 14% em 2017.

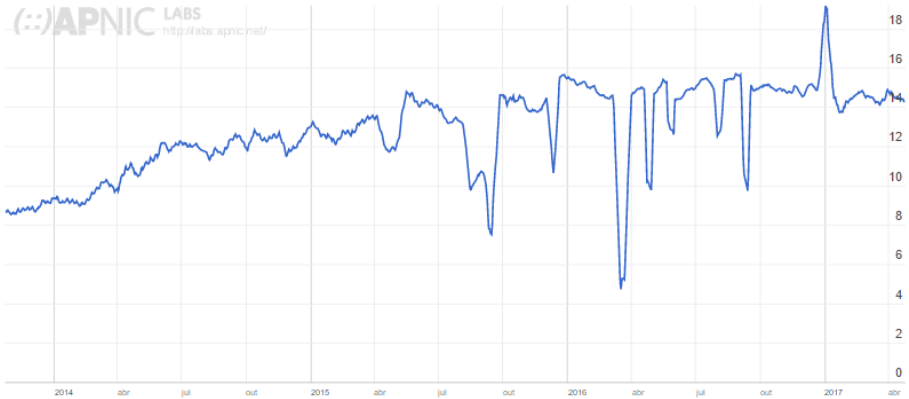


Figura 6.2: Crescimento global da taxa de utilização DNSSEC[40]

Estado da Assinatura DNSSEC

O processo de validação apenas é possível depois do processo de assinatura de zonas DNS estar concluído. Uma válida instalação dos serviços DNSSEC implica um processo correto de assinatura de todas as camadas hierárquicas do sistema DNS. Em 2010, a zona raiz foi assinada, realizando o primeiro passo para a implementação do DNSSEC.

A figura 6.3 representa o estado do processo de assinatura DNSSEC para uma parte dos domínios de topo, os domínios associados a códigos de países. A associação "Internet Society" identifica 5 fases na implementação do processo de assinatura DNSSEC [41]:

- Experimental: Experimentação interna anunciada ou observada;
- Anunciado: Comprometimento público para a implementação;
- Parcial: Zona assinada mas não está em operação (registo DS* não se encontra na raiz, ou seja o registo ainda não se encontra associado à cadeia de confiança global);
- Operacional: Zona assinada e operacional.

*Registo DS: Registo que contém a impressão digital (*hash*) da chave pública assinante (KSK) e permite determinar se uma zona tem o modo DNSSEC ativo.

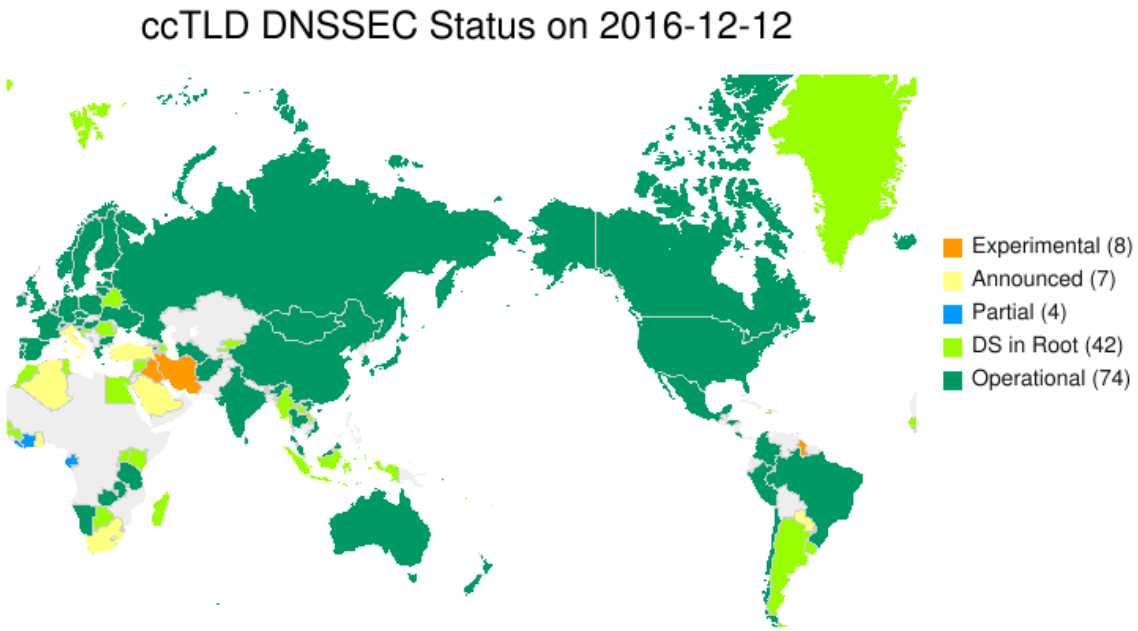


Figura 6.3: Estado de implementação DNSSEC ccTLDs (12/2016) [42]

Estado de Utilização DANE

A *Verisign Labs* disponibiliza uma estatística de implementação de registos TLSA em zonas DNS relatando que em maio de 2017 existiam cerca de 60 000 zonas DNS com registos TLSA associados. Em termos de crescimento, a mesma organização disponibiliza um gráfico, representado na figura 6.4, onde se verifica o contínuo aumento de zonas com registos TLSA, habilitando o protocolo DANE.

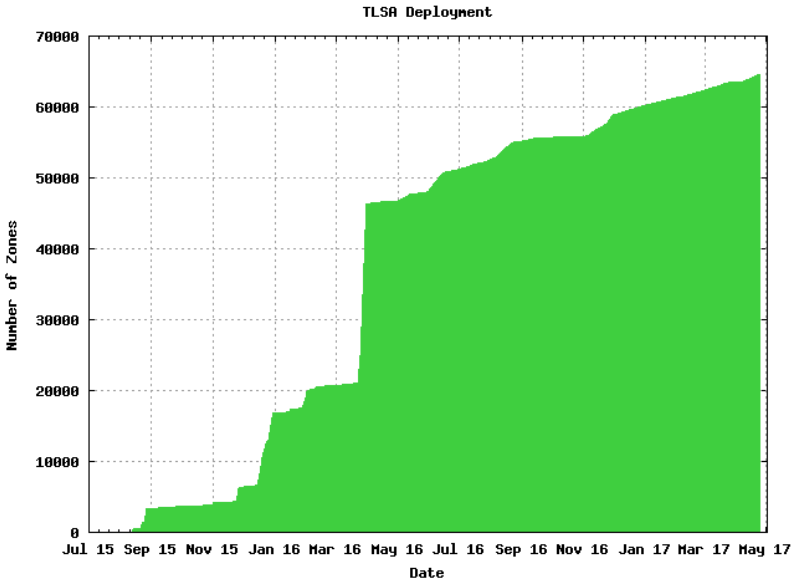


Figura 6.4: Evolução do número de zonas com registos TLSA associados [43]

Através das várias estatísticas disponíveis é possível verificar um aumento contínuo na utilização das tecnologias DNSSEC e DANE.

Quanto ao DNSSEC, o ato da assinatura da zona raiz DNS em 2010 constituiu um importante marco no que diz respeito ao incentivo à utilização DNSSEC, permitindo a sua implementação nas camadas subjacentes da hierarquia DNS. O facto de que uma enorme percentagem dos domínios de topo já providenciarem mecanismos de autenticação DNSSEC é uma prova de que a implementação está em crescimento, sendo possível projetar bons níveis de implementação a médio-longo prazo.

Quando ao DANE, a sua implementação tem vindo a aumentar mas o fato da sua implementação ser realizada sobre os mecanismos de DNSSEC faz com que os seus níveis de implementação dependam dos próprios níveis de implementação do DNSSEC, constituindo um entrave à sua utilização.

6.1.6 Estado de Implementação U.Porto

Utilizando o *add-on DNSSEC/TLSA Validator*[1] para o navegador *web Firefox*, é possível verificar se um determinado servidor *web* implementa mecanismos de segurança de DNSSEC e/ou DANE. Realizando um acesso à pagina *web* da U.Porto *sigarra.up.pt* foi possível confirmar que (ver figura 6.5) o domínio não utiliza técnicas de DNSSEC para assegurar comunicações seguras nos acessos ao serviço DNS. Como esperado o protocolo DANE também não se encontra implementado.

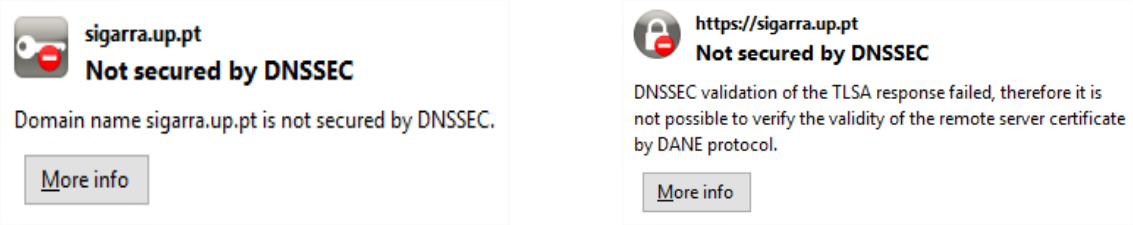


Figura 6.5: Informação sobre utilização de DNSSEC e DANE na U.Porto

Com o crescimento contínuo de implementações de DNSSEC, a sua futura implementação massiva é um acontecimento provável, e tratando-se a U.Porto de uma organização académica julga-se que faria sentido a implementação do sistema nas infraestruturas DNS pertencentes à Universidade do Porto. Este procedimento para além de propiciar a autenticação das comunicações do serviço DNS, serviria de base para uma implementação do protocolo DANE para os serviços da Universidade do Porto.

6.2 Convergence

Como já expresso na secção 2.3, o projeto foi apresentado por Moxie Marlinspike na palestra *SSL and the Future os Authenticity* realizada na conferência *Black Hat* em 2011, tratando-se de uma versão melhorada do projeto Perspectives [15].

De um ponto de vista de alto nível, a implementação Convergence pode ser descrita como um sistema distribuído capaz, em teoria, de suceder ao modelo atual de CAs através da utilização de uma comunidade de autoridades notárias que validam a legitimidade de um certificado digital, tendo o utilizador a palavra final sobre a validade do certificado em questão.

6.2.1 Funcionamento

O autor do projeto Convergence, em 2011, explicou que como forma a iniciar o projeto procedeu à identificação dos problemas cruciais do projeto Perspectives: plenitude, privacidade e capacidade de resposta [15]. Ei-los, de forma resumida:

[1] https://addons.mozilla.org/pt-PT/firefox/addon/dnssec-validator/

- Plenitude

  A abordagem Perspectives apenas funciona para a conexão inicial entre um navegador *web* e um servidor *web* seguro ou seja, todo o conteúdo de *background* (imagens, CSS, Javascript, etc) não é validado.

- Privacidade

  A necessidade da realização de uma consulta a uma autoridade notária depois de um acesso a um servidor *web* seguro faz com que exista, implicitamente, uma partilha da informação referente aos acessos dos utilizadores às autoridades notárias utilizadas.

- Capacidade de resposta

  As autoridades notárias ao realizarem cache dos certificados provocam problemas de sincronização nas respostas providenciadas às entidades finais. Quando uma entidade final consulta uma autoridade notária para a validação de um determinado certificado digital, essa autoridade realiza uma conexão ao servidor *web* em questão de modo averiguar o seu certificado digital. Seguidamente, a autoridade notária armazena a resposta em cache e inicia um processo periódico de acessos a esse servidor de modo a atualizar a informação sobre o certificado armazenado. O problema ocorre quando um servidor altera o seu certificado digital originando que a autoridade notária, no período de tempo em que ainda não realizou a atualização do certificado, responda a consultas com base no certificado desatualizado armazenado.

O projeto Convergence resolve os problemas identificados anteriormente através de um conjunto de novas implementações e alteração de processos.

- Privacidade

  O desafio dos problemas de privacidade é resolvido através de duas estratégias:

  A primeira consiste na permissão de que as entidades finais armazenem certificados localmente ou seja, no momento em que um certificado é considerado válido é realizado o seu armazenamento junto da entidade final de modo a que, num próximo acesso ao mesmo servidor *web* não seja necessário realizar de novo uma consulta às autoridades notárias. Este tipo de consultas passam a ser apenas necessárias quando se tratam de primeiros acessos a servidores *web* seguros ou quando o acesso é realizado após uma substituição do certificado digital por parte do servidor responsável.

  A segunda estratégia consiste na implementação do conceito de *notary bounce*. No momento em que uma entidade realiza uma consulta a um conjunto de autoridades notárias sobre a validade de um determinado certificado digital, é escolhida ao acaso uma dessas autoridades para ser a *notary bounce*. Esta autoridade funciona como um servidor *proxy* entre a entidade final e as restantes autoridades, não tendo visibilidade sobre as consultas realizadas. Deste modo, as autoridades notárias têm o conhecimento da informação que foi

solicitada mas não sabem quem a requisitou. Por outro lado, o servidor *proxy* sabe quem requisitou a informação mas não tem conhecimento da informação que foi solicitada.

- Capacidade de resposta

  A implementação resolve este problema por intermédio da alteração do processo de consulta às autoridades notárias. Neste método, para além de enviar o pedido de verificação aos servidores notários, as entidades finais enviam também o certificado proveniente do servidor *web* seguro de modo a validar esse certificado digital. Assim, as autoridades notárias apenas necessitam de realizar acessos aos servidores *web* quando o certificado enviado pelo cliente não corresponder ao certificado armazenado ou não existir nenhuma referência ao certificado em cache.

A implementação do Convergence, do lado do cliente consiste num *add-on* para o navegador *web Firefox*.

6.2.2 Pontos Fortes

De seguida apresentam-se os aspetos mais interessantes do projeto Convergence no que diz respeito ao foco deste documento:

- Agilidade de confiança

  Uma das maiores vantagens desta implementação consiste no controlo oferecido ao utilizador de modo a ser ele o responsável pelas suas definições de segurança, em termos da confiança imposta nas autoridades notárias. A capacidade em escolher quais servidores notários utilizar, a parametrização de um consenso entre eles e a possibilidade da gestão de um servidor notário por parte de qualquer entidade promove um sistema inserido num modelo de confiança flexível.

- Independência das CAs

  A possibilidade da não utilização de autoridades certificadoras para garantir acessos seguros através de certificados auto assinados consiste numa libertação dos utilizadores com o sistema atual. Através desta implementação, um detentor de um domínio pode auto-assinar o seu certificado, publicá-lo no servidor e esperar que os servidores notários o validem através dos próprios processos de verificação. Desta forma, a utilização de autoridades certificadoras para assegurar a autenticidade de uma correspondência entidade-chave pública tornar-se-ia secundária ou até desprezável.

- Migração nula para os servidores correntes

  A implementação do Convergence não requer qualquer alteração no modo de funcionamento dos servidores correntes. O sistema apenas prevê a criação dos vários tipos de servidores subjacentes à implementação do Convergence, e a implementação do protocolo no lado do cliente (navegadores *web*).

- Multiplicidade

  Os vários servidores notários podem utilizar diferentes protocolos de modo a validar um determinado certificado digital oferecendo assim ao utilizador a escolha de quais os métodos que considera mais seguros para a segurança das comunicações. Os diversos *back-ends* podem consistir na técnica utilizada pelo Perspectives, o DANE, o sistema atual de CAs, entre outras.

6.2.3 Pontos Fracos

Por outro lado, apresentam-se os aspetos mais inconvenientes de uma aposta na solução Convergence:

- Atraso de resposta

  A utilização de verificações extra a servidores de validação de certificados origina que as comunicações sofram de um atraso acrescido. Estes atrasos são minimizados pela solução através da implementação de mecanismos de cache, tanto do lado do cliente como no lado dos servidores notários. No entanto, numa verificação em que não haja a possibilidade de adquirir informação presente em cache, as comunicações extra originam um défice de capacidade de resposta ao sistema que afeta o seu desempenho.

- Necessidade de conhecimento

  Uma das maiores dificuldades do projeto consiste, curiosamente, num ponto considerado com uma vantagem do sistema. A responsabilidade providenciada ao utilizador no que diz respeito às suas relações de confiança e por consequência à segurança das suas comunicações prevê uma premissa de que os utilizadores conseguem tomar decisões conscientes e adequadas às suas necessidades.

  Mas com toda a certeza, a maioria dos utilizadores, devido à falta de conhecimento geral sobre os aspetos de segurança na Internet, utilizaria as configurações base do sistema, não usufruindo dos serviços de segurança acrescentada pela implementação.

- Problema *Citibank*

  Esta adversidade surge pelo fato de algumas páginas *web*, como a *citibank.com*, servirem-se de uma vasta quantidade de certificados SSL alternativos para o mesmo domínio. Isto origina que, utilizando o sistema Convergence, num processo de validação de um certificado proveniente de uma página deste tipo, o utilizador receberia o certificado da página e enviaria esse certificado numa consulta a um conjunto de servidores notários. Esses servidores procederiam então à obtenção do certificado do servidor através da sua própria perspetiva, de modo a comparar com o certificado entregue pelo cliente. Como a página tem um conjunto de certificados SSL válidos, os notários receberiam certificados diferentes uns dos outros e do certificado enviado pelo cliente, tornando impossível uma validação concreta.

- Portais cativos

  Um portal cativo, em inglês *captive portal*, consiste num *software* que controla o acesso à Internet em redes públicas. Este conceito consiste num problema para a solução Convergence devido à limitação imposta do acesso à Internet, impedindo o acesso dos utilizadores aos servidores notários para realizar validações de certificados das páginas às quais o *software* permite acesso.

6.2.4 Estado de Utilização

O projeto encontra-se disponível para implementação do lado do cliente como um *add-on* do navegador *web Firefox*.

O sistema acabou por não ter sucesso nos aspetos relativos à sua instalação e implementação massiva devido principalmente às restrições impostas pelos navegadores relativamente às decisões de confiança e à falta de conhecimentos dos utilizadores para a realização de configurações personalizadas permitidas pelo sistema [44].

6.2.5 Estado de Implementação U.Porto

Não existe qualquer informação referente à implementação da solução Convergence por parte da U.Porto.

6.3 Certificate Transparency

A solução Certificate Transparency (CT), proposta pela Google, consiste numa otimização do sistema atual de distribuição de chaves públicas e tem como principal objetivo eliminar a vulnerabilidade referente à possibilidade das diferentes CAs emitirem certificados SSL sem a autorização dos domínios em questão. Os benefícios desta otimização são proporcionados através de um conjunto de operações de monitorização e auditoria dos certificados emitidos[18].

De modo a possibilitar a monitorização de certificados, a solução implementa uma estrutura constituída por três componentes específicos: servidores de registos (*logs*), monitores e auditores.

Os conceitos base desta solução encontram-se explicitados na subsecção 2.3.5 deste documento.

6.3.1 Funcionamento

6.3.1.1 Servidores de Registo

Como explicitado anteriormente, os servidores de registos consistem em servidores que têm como principais funções armazenar certificados SSL e permitir sua auditoria pública.

De modo a cumprir os requisitos técnicos para as suas funções, os servidores de registo devem deter as seguintes propriedades:

- Estrutura *append-only*: a sua estrutura deve obedecer a uma organização onde apenas podem ser adicionados registos à estrutura, sendo proibidas atividades de edição ou remoção;
- Utilização de *Hash Merkle Tree*: os servidores devem utilizar o mecanismo criptográfico de *hash Merkle Tree* de modo a prevenir e identificar alterações aos seus registos, e de proporcionar as funções da estrutura *append-only*;
- Auditoria pública: os registos dos servidores devem estar disponíveis para que todas as entidades os possam consultar e realizar verificações.

Quando uma qualquer entidade submete para registo um certificado válido, o servidor de registo responde com um SCT (*Signed Certificate Timestamp*) que consiste numa promessa de inclusão do certificado no servidor num determinado período de tempo, MMD (*Maximum Merge Delay*). Um servidor TLS deve entregar o SCT juntamente com o certificado durante o protocolo de iniciação de conexão [45].

A solução suporta três métodos de entrega do SCT com o certificado:

- Extensão X509v3

  Inclusão do SCT nas extensões do certificado SSL x509v3. O ciclo de criação/utilização do certificado inicia com a emissão do certificado não final por parte de uma CA, seguida de um pedido de inclusão num servidor de registo. O servidor de registo procede à entrega do SCT, o que permite à CA adicionar o SCT às extensões do certificado, e assim concluir o processo de criação do certificado. Terminada a criação do certificado, a CA entrega-o ao servidor TLS em questão.

- Extensão TLS

  Neste método, o servidor TLS entrega o SCT por intermédio da comunicação TLS com o cliente. Uma CA emite um certificado SSL e entrega-o ao servidor TLS que o requisitou. Seguidamente, o servidor TLS realiza o pedido de inclusão num servidor de registos de modo a obter o SCT para o certificado. Depois da obtenção do SCT, o servidor TLS inclui o SCT na comunicação TLS quando um cliente realizar uma conexão ao seu servidor.

- OCSP *Stapling*

  Entrega realizada através de operações de OCSP *Stapling*. Uma CA, no momento em que entrega um certificado SSL emitido por si a um servidor TLS, faz um pedido de inclusão desse certificado a um servidor de registo, ao que este responde com o SCT. Desta forma, o servidor TLS pode realizar uma consulta OCSP à CA, referente ao SCT associado ao seu certificado de modo a incluí-lo na comunicação TLS com um cliente.

Merkle Hash Tree

Uma *Merke Hash Tree* adaptada à solução CT consiste numa árvore binária constituída pelos elementos: *leaf hashs*, *nós hash*, *hash raiz* e certificados SSL [46].

- Um *leaf hash* consiste na *hash* de um certificado SSL armazenado pelo servidor de registo;
- Um *nó hash* consiste na *hash* de um par de elementos situados diretamente abaixo na estrutura hierárquica do servidor, podendo consistir na *hash* de um par de *leaf hashs* ou na *hash* de um par de *nós hashs*;
- O nó raiz, reconhecido por *Hash Merkle Tree*, representa o nó situado no topo da hierarquia e consiste no nó do qual o seu valor depende de todos os nós existentes na estrutura;
- Os certificados SSL consistem nos certificados submetidos nos servidores de registo.

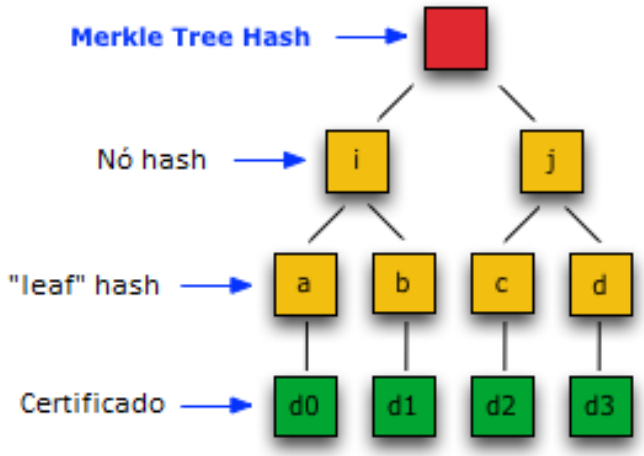


Figura 6.6: Estrutura de uma *Hash Merkle Tree*[46]

Uma estrutura deste tipo permite provar que todos os certificados foram adicionados corretamente ao servidor e que um determinado certificado foi adicionado ao servidor. Estas competências são proporcionadas através de duas provas criptográficas: a prova de consistência *Merkle* e a prova de auditoria *Merkle*.

Prova de consistência *Merkle*

Com o contínuo aumento do número de registos nos servidores, este tipo de prova permite verificar que a versão mais recente de um determinado servidor de registos é consistente, ou seja, que a nova versão inclui toda a informação da versão anterior e que toda a nova informação vem após a informação da versão anterior. A consistência de um servidor de registo permite provar que:

- Nenhum certificado foi datado com uma data anterior e inserido no servidor;
- Nenhum certificado foi modificado.

De maneira a provar que a nova versão inclui toda a informação anterior é necessário que a versão anterior se trate de um subconjunto da nova versão, e para provar que toda a nova informação vem de forma posterior à informação da versão anterior, é necessário verificar que a nova versão consiste na concatenação da versão anterior com a nova informação. A prova de consistência consiste no mínimo conjunto de nós necessários para satisfazer as duas provas anteriores. Se o nó raiz calculado consistir no mesmo valor apresentado pelo servidor, então é possível confirmar a consistência do servidor.

Monitores e auditores utilizam provas de consistência para verificar o funcionamento correto de um servidor de registo.

Prova de auditoria *Merkle*

Este tipo de prova permite verificar se um determinado certificado foi incluído num servidor de registo. Mediante a estrutura, a prova consiste nos *nós hash* necessários para calcular todos os valores entre o *leaf hash* do certificado em questão e o nó raiz. Se o nó raiz calculado consistir no mesmo valor apresentado pelo servidor, então é possível concluir que o certificado existe nesse mesmo servidor.

Este tipo de prova é especialmente importante devido à imposição do sistema referente à obrigatoriedade de qualquer certificado SSL existir num servidor de registo.

Estas provas permitem prover aos vários auditores formas de garantir as condições necessárias para o bom funcionamento do sistema, sem ser necessário deterem toda a informação dos servidores de registos. De forma a realizar as verificações de consistência e auditoria, os auditores apenas recebem dos servidores as provas, que consistem em *hashs* de nós intermediários, e através delas derivam a *hash* do nó raiz. Quanto aos monitores, estas entidades descarregam periodicamente toda a informação armazenada nos servidores de registos e calculam as provas de modo a realizar as verificações.

6.3.1.2 Monitores

Os monitores têm como funções a realização de procuras de situações suspeitas nos servidores de registo, tais como certificados não legítimos, extensões de certificados anormais, e o descarregamento de todos os novos registos de um servidor de registos, de forma a terem, periodicamente, uma imagem completa da informação dos servidores de registo.

6.3.1.3 Auditores

Os auditores têm como principais funções verificar a integridade dos servidores de registo através das provas de consistência e a consulta da existência de certificados nos servidores através das provas de auditoria.

Os monitores e os auditores estabelecem um canal de comunicação entre si de modo a verificar que a sua própria perspetiva de um determinado servidor de registo é consistente com a perspetiva das outras entidades.

Neste sistema, os clientes TLS, no protocolo de iniciação de conexão com um servidor, devem verificar o SCT recebido, validando a assinatura do servidor de registo para averiguar se este foi emitido por um servidor de registo válido e que foi emitido para o certificado em questão. Se existirem inconsistências na validade do SCT, o cliente TLS deve rejeitar o certificado associado.

6.3.2 Pontos Fortes

De seguida apresentam-se os aspetos mais interessantes do projeto CT no que diz respeito ao foco deste documento:

- Maior controlo sobre a emissão de certificados

  Uns dos propósitos principais da solução consiste na eliminação ou atenuação da vulnerabilidade reconhecida no sistema atual que consiste na capacidade de qualquer entidade certificadora em emitir certificados sem a autorização dos domínios em questão. Para isso, o sistema exige a submissão de todos os certificados emitidos em servidores de registos e disponibiliza a informação desses servidores de forma a que qualquer entidade possa realizar consultas e verificações.

- Maior fiscalização sobre o sistema SSL

  A existência da possibilidade da realização de verificações e consultas aos servidores que contêm a informação referente aos certificados por parte de qualquer entidade pública promove uma fiscalização mais alargada sobre todo o sistema de distribuição de chaves criptográficas públicas.

- Maior segurança

  Através da possibilidade de monitorização e auditoria dos servidores de registo e dos seus certificados, a solução permite uma mais rápida deteção de certificados ilegítimos e, por consequência, providencia uma maior segurança ao nível da legitimidade dos certificados SSL.

6.3.3 Pontos Fracos

Por outro lado, apresentam-se os aspetos mais inconvenientes de uma aposta na solução CT:

- Dependência de entidades certificadoras

  A solução não prevê uma diminuição da dependência às entidades certificadoras para o processo de distribuição de chaves públicas, o que proporciona que alguns dos problemas relativos à sua utilização sejam herdados pela solução.

- Impacto na infraestrutura existente

  As mudanças mais relevantes associadas à implantação da otimização consistem na criação e utilização de servidores de registo para armazenar os certificados emitidos, a alteração nos fluxos relacionados com os certificados de modo a integrar os SCT e a criação de componentes para monitorização e auditoria.

- Aumento de pontos de falha na infraestrutura

  O aumento no número de componentes aumenta, intrinsecamente, o número de pontos de possível falha na infraestrutura. Qualquer componente inserido no sistema é passível de falhas e de ataques o que aumenta a probabilidade de colapso do sistema ou de partes fundamentais do mesmo.

6.3.4 Estado de Utilização

Sendo um projeto da Google, a implementação e utilização desta solução foca-se especialmente em ambientes da organização. Em maio de 2015, a Google informou a obrigatoriedade da utilização da solução CT para todos os certificados do tipo EV, emitidos a partir de janeiro de 2015. A não aplicação da solução por parte dos emissores/detentores do certificado significa o não reconhecimento da validade desse certificado pelo navegador *web*.

Numa afirmação do reconhecimento e da evolução da técnica, a Google anunciou a obrigatoriedade da utilização da abordagem Certificate Transparency para a aceitação de novos certificados pelo navegador Chome, a ter efeito a partir de Outubro de 2017 [47].

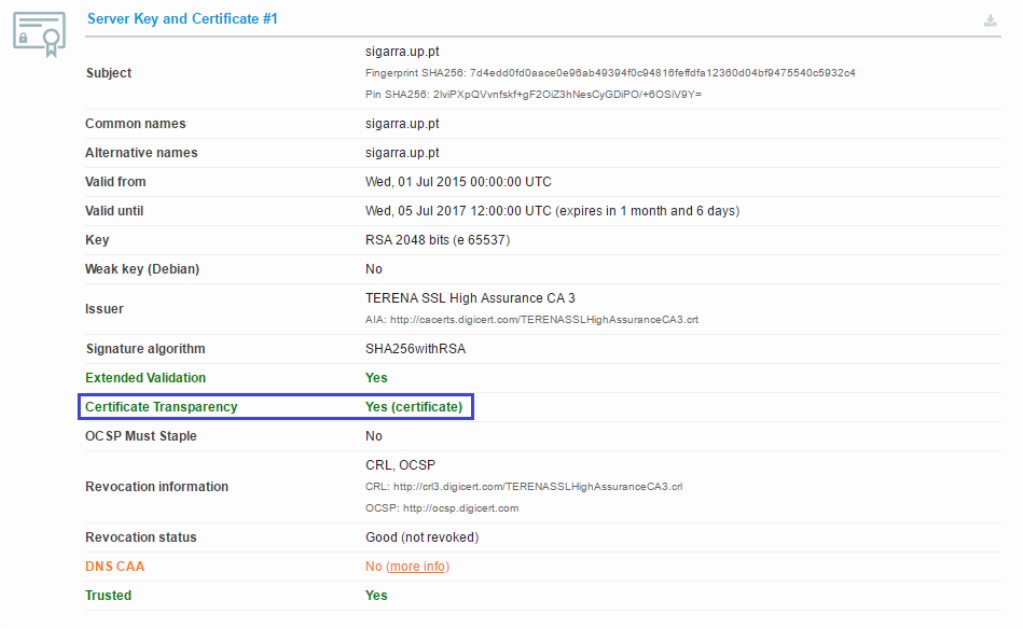
Relativamente à implementação de servidores de registo, atualmente existe um vasto número de servidores desse género associados a vários operadores [48].

6.3.5 Estado de Implementação U.Porto

Por intermédio da ferramenta *web SSL Server Test* da *Qualys SSL Labs*, é possível verificar os diferentes mecanismos de segurança que um determinado servidor *web* utiliza.

Quanto ao servidor *web sigarra.up.pt* referente à página *online* da U.Porto, a ferramenta demonstra que, tal como ilustra a figura 6.7, o certificado associado utiliza mecanismos da solução CT.

A prova de que o certificado utiliza mecanismos referentes à solução CT também pode ser comprovado através da consulta dos detalhes do certificado SSL correspondente ao domínio *sigarra.up.pt*, ilustrado na figura 6.8, onde existe o campo *1.3.6.1.4.1.11129.2.4.2* nas extensões do mesmo, referente ao SCT providenciado pelo servidor de registo.



Server Key and Certificate #1

Subject	sigarra.up.pt Fingerprint SHA256: 7d4edd0fd0asce0e98ab49394f0c94816feffdfa12360d04bf9475540c5932c4 Pin SHA256: 2lviPXpQVvnfskf+gF2OiZ3hNesCyGDiPO/+6OSiV9Y=
Common names	sigarra.up.pt
Alternative names	sigarra.up.pt
Valid from	Wed, 01 Jul 2015 00:00:00 UTC
Valid until	Wed, 05 Jul 2017 12:00:00 UTC (expires in 1 month and 6 days)
Key	RSA 2048 bits (e 65537)
Weak key (Debian)	No
Issuer	TERENA SSL High Assurance CA 3 AIA: http://cacerts.digicert.com/TERENASSLHighAssuranceCA3.crt
Signature algorithm	SHA256withRSA
Extended Validation	Yes
Certificate Transparency	Yes (certificate)
OCSP Must Staple	No
Revocation information	CRL, OCSP CRL: http://crl3.digicert.com/TERENASSLHighAssuranceCA3.crl OCSP: http://ocsp.digicert.com
Revocation status	Good (not revoked)
DNS CAA	No (more info)
Trusted	Yes

Figura 6.7: Informação *SSL Server Test*: domínio *sigarra.up.pt*[49]

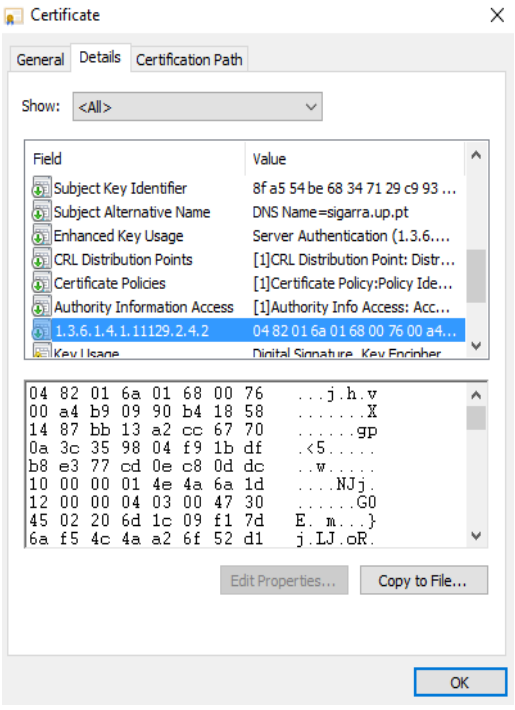


Figura 6.8: Certificado *sigarra.up.pt* - extensão relativa ao SCT

6.4 Considerações

Neste capítulo efetuou-se uma análise objetiva das três principais otimizações/alternativas encontradas de modo a evidenciar uma delas para a realização de uma prova de conceito e de um conjunto de diretivas de instalação para a sua integração na U.Porto.

De entre as três opções, a escolhida foi a solução DANE pelo seguinte conjunto de razões:

- Acréscimo de segurança relativo ao sistema atual;
- Capacidade de independência face às CAs;
- Contínua evolução do sua utilização;
- Capacidade de adaptação ao sistema atual.

Capítulo 7

Proposta

Neste capítulo apresenta-se uma descrição da solução proposta através de uma prova de conceito do seu funcionamento e expõem-se diretivas de implementação da solução orientadas à U.Porto. Com o intuito de prover uma análise ao valor da solução, é apresentado um conjunto de considerações que permite estabelecer uma relação de comparação com o sistema atual de distribuição de chaves públicas.

7.1 Pré-considerações

Nesta secção do documento são enumerados e definidos os principais aspetos que permitem avaliar a validade de uma solução que seja capaz de substituir ou diminuir o papel das Entidades Certificadoras na infraestrutura de chaves públicas utilizada na Internet.

Segurança

Garantias de segurança providenciadas pelos mecanismos utilizados pela solução. São tidos em consideração os aspetos relativos à segurança das comunicações dos utilizadores, segurança das infraestruturas utilizadas, capacidade de resistência a vários tipos de ataques, etc.

Privacidade

A privacidade do utilizador deve ser protegida pelas diversas práticas empregues pela solução.

Flexibilidade

Métodos que proporcionam aos utilizadores controlo sobre as relações de confiança existentes, nomeadamente na capacidade de revogação de confiança, existência de alternativas relativas ao estabelecimento de relações de confiança, etc.

Escalabilidade

O modelo adotado pela proposta deverá consistir num sistema que possibilite um crescimento sustentado sem notórias perdas de desempenho mantendo, sobretudo, as características responsáveis pela segurança e desempenho do modelo.

Disponibilidade

Capacidade do sistema em oferecer os seus serviços com um nível de desempenho adequado, independentemente da situação.

Latência

A interatividade com o utilizador deve respeitar um período de tempo aceitável de modo a que a duração das comunicações entre as várias entidades do modelo sejam o menos inconvenientes possíveis para a experiência de utilização.

Usabilidade

Do ponto de vista do utilizador, o sistema deverá ser o mais simples possível, possibilitando a sua aplicação pelo maior número possível de utilizadores.

Custos de transição

Custos referentes às alterações de infraestruturas e *software* necessárias ao modelo atual para a implementação do sistema. Altos custos de transição originam pouca motivação por parte das diversas organizações em investir no sistema.

Como forma de sustentação e de referência à proposta a elaborar, seguidamente é apresentada uma análise ao sistema atual de infraestrutura de chaves públicas através dos aspetos identificados.

- Segurança

  A existência de inúmeras CAs origina uma maior probabilidade de ataques bem-sucedidos a estas entidades, devido ao elevado número de casos onde existe falta de mecanismos satisfatórios de segurança. A existência de casos reais de ataques bem-sucedidos a CAs de alto nível constitui uma prova da debilidade do sistema.

  De uma outra perspetiva, o monopólio criado pelo restrito conjunto de entidades certificadoras que controlam o sistema, aliado à falta de alternativas para o estabelecimento de relações de confiança origina uma dependência direta dos utilizadores a este tipo de entidades. Devido a este fator, os utilizadores não têm forma de comunicar com segurança com um domínio cujo certificado seja emitido por uma entidade certificadora que não confiam.

- Privacidade

  A necessidade dos utilizadores em realizar consultas a servidores OCSP para a verificação do estado de revogação dos certificados descarregados consiste num dos maiores problemas de privacidade do sistema devido à exposição dos acessos aos domínios que os utilizadores pretendem aceder. A solução providenciada pelo sistema para o problema de privacidade OCSP consiste num outro método de verificação do estado de revogação denominado OCSP *Stapling* que permite eliminar os acessos a entidades exteriores à comunicação.

- Flexibilidade

  No atual sistema de entidades certificadoras, os utilizadores não têm alternativa à imposição de relações de confiança para com CAs impostas pelo modelo. Se um utilizador não confiar numa CA pode, com alguns conhecimentos técnicos, excluir essa entidade da sua base de dados de entidades confiáveis com a consequência de que todos os domínios certificados por essa CA surgirem como não confiáveis pelo navegador *web*. O modelo carece de uma forma de permitir aos utilizadores revogar seletivamente relações de confiança com CAs, sem o prejuízo de uma utilização segura em outras comunicações.

- Escalabilidade

  O sistema atual tem a capacidade de estender as suas funcionalidades para um maior número de utilizadores. Naturalmente, o aumento de utilizadores fomenta um maior número de pontos de falha do modelo devido à necessidade de criação de mais elementos passiveis de ataques e falhas, tais como entidades certificadoras intermediárias, servidores de verificação de estados de revogação, etc.

- Disponibilidade

  A existência de servidores OCSP e servidores que alojam listas de certificados revogados consiste na principal ameaça à disponibilidade da funcionalidade do sistema. Todo o sistema gira em torno da validade dos certificados digitais providenciados pelos servidores aos utilizadores. A impossibilidade de um utilizador poder verificar a validade de um certificado digital descarregado através de uma comunicação com um servidor ocasiona uma falha ao sistema no que respeita à sua disponibilidade.

- Latência

  O sistema não prevê operações dispendiosas relativamente ao tempo de execução. Os utilizadores, através da comunicação com o servidor descarregam o certificado correspondente ao domínio ao qual pretendem aceder e realizam um conjunto de verificações ao certificado. De todas as verificações realizadas pelo utilizador a consulta a um servidor OCSP e/ou a uma lista de certificados revogados consiste na que pode originar mais atraso devido ao acesso externo necessário. A utilização de OCSP *Stapling* permite simplificar ainda mais o processo de verificação do lado do utilizador eliminando a necessidade de acessos a servidores OCSP e assim suprimir o atraso provocado por estes.

- Usabilidade

  No que diz respeito à usabilidade, do ponto de vista do utilizador o sistema atual é extremamente simples muito devido à falta de transparência nas condições de confiança impostas.

- Custos de transição

  Não se aplica.

7.2 DANE

Tendo em vista uma proposta para a implementação da especificação DANE na U.Porto, seguidamente apresenta-se uma prova de conceito realizada numa rede interna composta por máquinas virtuais de modo a descrever a sua instalação e provar o seu funcionamento.

7.2.1 Prova de Conceito

A prova de conceito completada consistiu na instalação e implementação dos vários componentes necessários para colocar em funcionamento a especificação DANE. Nesta secção realiza-se uma descrição da rede utilizada, seguida das várias fases necessárias para alcançar o estado final da implementação.

7.2.1.1 Rede

Para a realização da prova de conceito relativa à implementação da solução DANE, foi necessário configurar uma rede composta por máquinas virtuais interligadas. Nesta estrutura, cada máquina é responsável por um conjunto de ações que combinadas entre si permitem provar o funcionamento da tecnologia DANE numa rede *intranet* simples. A figura 7.1 representa a estrutura da rede, composta por:

- *Debian Root, Debian 1*: Máquinas responsáveis pela gestão dos certificados existentes na rede;
- *Debian 4*: Máquina onde se encontra alojado o servidor DNS recursivo e o navegador *web* utilizado para consultar o domínio criado para teste, *www.exemplo.fms*.
- *Debian 5*: Máquina onde se encontra alojado o servidor DNS autoritário responsável pela zona criada, *exemplo.fms*;
- *Debian 6*: Máquina responsável por alojar o servidor *web* onde se encontra a página HTTP de teste.

As verificações das comunicações foram realizadas através da ferramenta *Wireshark*, que permite inspecionar as diversas trocas de informações entre os vários componentes da rede e assim validar a implementação da tecnologia.

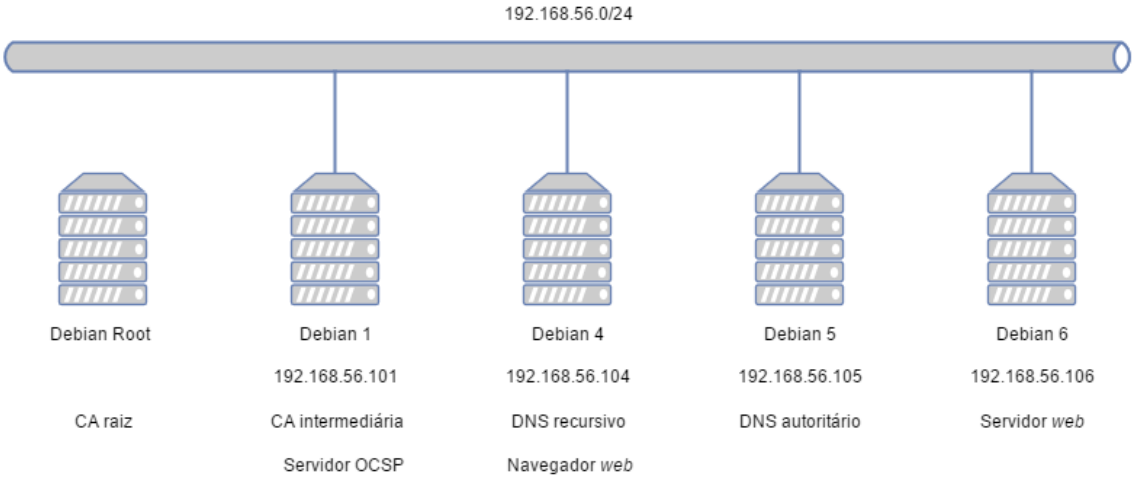


Figura 7.1: Estrutura da rede - prova de conceito DANE

Seguidamente, são expostas as diferentes fases utilizadas na implementação DANE realizada. Os detalhes das diferentes fases de configuração encontram-se documentados no Anexo B deste documento.

7.2.1.2 Fase 1: Servidores DNS

Com o intuito de providenciar um mecanismo de tradução de nomes de máquinas em endereços de IP foram configurados dois servidores DNS alojados nas máquinas *Debian 4* e *Debian 5*. Para ambas foi instalada a ferramenta *bind9* para o sistema operativo *debian*.

Nesta prova, o servidor recursivo consiste na máquina responsável pela comunicação DNS com o cliente, constituindo uma ponte entre a informação armazenada no servidor autoritário e o cliente que pretende obter essa informação. O servidor autoritário tem como principal função providenciar respostas a consultas provenientes do servidor recursivo.

7.2.1.3 Fase 2: Servidores DNS com DNSSEC

Numa segunda fase, recorreu-se ao pacote DNSSEC-TOOLS[1] de modo a gerar a implementação DNSSEC para a zona criada anteriormente. Especificamente, foi utilizada a ferramenta "*zonesigner*", de modo a gerar os registos associados à implementação DNSSEC necessários a partir dos existentes na zona.

Depois de completo o processo de assinatura dos vários registos no servidor autoritário foi necessário recorrer a uma indicação da chave KSK desse mesmo servidor, no servidor recursivo. Este procedimento tem como objetivo conceder uma forma de o servidor conseguir autenticar as mensagens provenientes do servidor autoritário, visto que as mensagens são assinadas digitalmente pelo emissor.

[1] https://www.dnssec-tools.org/

Este processo consiste numa operação análoga ao processo referente a um sistema *online* onde, como explicado na subsecção 2.3.1 deste documento, a chave da raiz do sistema KSK é obtida pelo cliente através de um outro método que não o DNS.

7.2.1.4 Fase 3: Servidor HTTP

Depois da configuração DNSSEC, o passo seguinte consistiu na configuração de um servidor HTTP e da criação de uma página de teste denominada *www.exemplo.fms* através da ferramenta *Apache*. Tendo em conta o requisito referente à existência do protocolo TLS nos servidores *web* da rede para a implementação da solução DANE, procedeu-se à emissão de um certificado digital para o servidor criado. Para a geração do certificado, foi utilizada a CA intermediária concebida anteriormente para o teste de implementação *openSSL*, especificado no capítulo 4. Esta etapa culmina com a implementação do certificado no servidor, de modo a ser transmitido nas comunicações com os clientes.

7.2.1.5 Fase 4: Registo TLSA

A última etapa consiste na criação e implementação do registo TLSA correspondente ao servidor *web* no servidor DNS autoritário da rede. Para a geração do registo utilizou-se um serviço *web* que permite a criação de um registo TLSA através da escolha de um conjunto de parâmetros de utilização e a indicação do certificado em questão. A implementação do registo TLSA consiste na inclusão do registo no servidor DNS autoritário seguida da execução da ferramenta *zonesigner* de modo a atualizar e/ou criar os registos necessários para a implementação DNSSEC.

Registo TLSA implementado:

```
_443._tcp.www.exemplo.fms.  IN TLSA 1 0 1
741aaa3c8129793a6abf8d2061c421d6101c43eb8b04f65d86c9b93ad2d9bdea
```

7.2.1.6 Verificação

Primeiramente, através da ferramenta *Wireshark*, foram analisadas as várias trocas de informação no acesso ao servidor *web* criado, mais concretamente à página *www.exemplo.fms*. O intuito da análise consiste na verificação da permuta de mensagens correspondentes aos protocolos DNSSEC e DANE, para provar a possibilidade de uma validação do tipo DANE ao domínio em questão. A análise *Wireshark* encontra-se detalhada na secção B.5 referente ao anexo B deste documento.

De forma a realizar a validação DANE foi necessário inspecionar os parâmetros do registo TLSA obtido por meio da comunicação com o servidor DNS para determinar de que maneira a informação entregue pelo servidor *web* deve ser validada. De seguida apresentam-se os parâmetros do registo TLSA juntamente com a sua definição.

- Utilização: 1

  Especificação de que a informação presente no registo TLSA, correspondente ao certificado, deverá corresponder ao certificado entregue pelo servidor responsável pela comunicação TLS e que o certificado deve ser validado através do esquema PKI de CAs.

- Seletor: 0

  Especificação de que, depois da entrega por parte do servidor TLS, deve ser utilizado o certificado completo para a comparação com a informação presente no campo de dados do registo TLSA.

- Tipo de correspondência: 1

  Especificação do algoritmo SHA-256 para a comparação entre o campo de dados correspondente à informação do certificado no registo TLSA e o certificado entregue pelo servidor TLS.

Depois da análise do registo TLSA, concluiu-se que se deve comparar a *hash* SHA-256 do certificado completo entregue pelo servidor TLS com o campo de dados correspondente à informação do certificado no registo TLSA.

Através da ferramenta *Wireshark* foi possível obter o certificado enviado pelo servidor *web* ao utilizador na comunicação TLS. O ficheiro `exemplo.fms.pem` consiste na representação no formato pem do certificado em questão.

De modo a verificar a impressão digital (*hash*) SHA-256 do certificado realizou-se o seguinte comando:

```
#openssl x509 -noout -fingerprint -sha256 -inform pem -in exemplo.fms.pem
```

Resultado:

```
SHA256 Fingerprint=74:1A:AA:3C:81:29:79:3A:6A:BF:8D:
20:61:C4:21:D6:10:1C:43:EB:8B:04:F6:5D:86:C9:B9:3A:D2:D9:BD:EA
```

Informação obtida no registo TLSA:

```
741aaa3c8129793a6abf8d 2061c421d6101c43eb8b04f65d86c9b93ad2d9bdea
```

Como é possível verificar, a informação coincide.

Este tipo de verificações e validações podem ser realizadas pelas aplicações do lado do utilizador final de modo a informar a validade de um determinado certificado entregue numa comunicação com um servidor.

7.2.2 Diretivas de Implementação

A implementação DANE numa infraestrutura como a da U.Porto compreende um curto conjunto de passos relacionados com as premissas da solução.

A operação mais importante no que diz respeito à instalação DANE consiste na implementação DNSSEC no serviço DNS da U.Porto. Este procedimento serve de base para as várias possibilidades de segurança oferecidas pelo DANE.

Através de um serviço *web* de análise DNSSEC disponibilizado pela *Verisign Labs*[2] foi possível verificar o estado atual da implementação DNSSEC no domínio *up.pt*. Este serviço indica a condição DNSSEC das várias zonas constituintes do domínio. A figura 7.2 representa o resultado da consulta ao domínio *sigarra.up.pt*, onde se constata que tanto na zona raiz, como na zona pt o estado da implementação DNSSEC encontra-se pronto a utilizar, contrastando com o domínio up.pt.

.	✓ Found 2 DNSKEY records for . ✓ DS=19036/SHA-256 verifies DNSKEY=19036/SEP ✓ Found 1 RRSIGs over DNSKEY RRset ✓ RRSIG=19036 and DNSKEY=19036/SEP verifies the DNSKEY RRset
pt	✓ Found 2 DS records for pt in the . zone ✓ Found 1 RRSIGs over DS RRset ✓ RRSIG=14796 and DNSKEY=14796 verifies the DS RRset ✓ Found 2 DNSKEY records for pt ✓ DS=18303/SHA-1 verifies DNSKEY=18303/SEP ✓ Found 2 RRSIGs over DNSKEY RRset ✓ RRSIG=18303 and DNSKEY=18303/SEP verifies the DNSKEY RRset
up.pt	✗ No DS records found for up.pt in the pt zone ✗ No DNSKEY records found ✓ sigarra.up.pt A RR has value 193.137.35.140 ✗ No RRSIGs found

Figura 7.2: Estado de implementação DNSSEC U.Porto

Os aspetos referidos provam a necessidade da implementação DNSSEC na zona *up.pt*, que constitui a primeira diretiva de implementação DANE.

Quanto à instalação para acessos HTTP, as diretivas de implementação centram-se em dois aspetos:

- Inclusão de registos TLSA referentes aos domínios *web* nos servidores DNS responsáveis;
- Providenciar informação aos utilizadores de forma a instalar um serviço de verificação DANE nos seus navegadores *web*. Como referido numa etapa anterior do documento, existem *add-ons* gratuitos para os principais navegadores *web* que realizam validação DANE, como por exemplo o *DNSSEC/TLSA Validator*.

[2] http://dnssec-debugger.verisignlabs.com/

Quanto à instalação para comunicações via SMTP, as diretivas de implementação consistem na:

- Inclusão de registos TLSA referentes aos servidores SMTP nos servidores DNS responsáveis;
- Ativação do modo de validação DANE nos servidores SMTP da infraestrutura.

7.3 Considerações

Como termo de comparação com a análise realizada para o sistema atual, apresenta-se uma análise à solução DANE através dos aspetos identificados no início do presente capítulo.

- Segurança

  O sistema DANE permite a continuação da utilização do sistema de entidades certificadoras adicionando uma camada adicional de segurança que consiste na indicação, por parte dos detentores dos domínios, do certificado que os seus utilizadores devem aplicar. Através desta indicação a solução não limita a capacidade das CAs em emitir um certificado para um domínio sem a sua prévia autorização, mas confere ao utilizador uma forma de verificar a validade de um certificado através da indicação do detentor do domínio.

  No que diz respeito ao protocolo SMTP a solução, para além de adicionar a camada de segurança associada à referência do certificado a utilizar, confere um acréscimo da segurança eliminando a possibilidade de ataques de *downgrade*, obrigando as comunicações a serem seguras quando existe essa possibilidade.

- Privacidade

  As questões de privacidade permanecem inalteradas se a implementação da solução compreender a continuação da utilização de CAs. Se for utilizada a variante relativa aos certificados auto-assinados, não existe a necessidade de consultas OCSP e por consequência os problemas de privacidade associados a essas consultas são eliminados.

- Flexibilidade

  A inclusão de uma nova camada de validação do certificado digital e a possibilidade da utilização de certificados auto-assinados correspondem a um alargamento da flexibilidade do sistema.

  A utilização de certificados auto-assinados permite a total independência dos serviços das CAs, responsáveis por uma boa quota da inflexibilidade do sistema atual, mas o sistema continua com a desvantagem referente à dependência das entidades que gerem o serviço DNS.

  De uma forma geral, o sistema confere aos utilizadores uma gama mais alargada de hipóteses de verificar a associação entre uma entidade e a sua chave criptográfica pública.

- Escalabilidade

  A solução DANE permite uma escalabilidade idêntica ao sistema atual de distribuição de chaves públicas. Os pontos adicionados pela tecnologia não representam uma barreira no que diz respeito ao aumento do número de utilizadores nem de serviços.

- Disponibilidade

  A introdução de novos tipos de registos DNS e a sua correta implementação representa a principal ameaça aos aspetos relativos à disponibilidade oferecida pela solução. Uma configuração imprópria, tanto do DNSSEC como dos registos TLSA, representarão uma quebra do serviço e a sua consequente indisponibilidade.

- Latência

  O DANE exige um período referente à obtenção do registo do certificado através do serviço DNS seguro por DNSSEC e um intervalo de tempo correspondente à validação desse registo em comparação com o certificado apresentado pelo servidor.

  Comparativamente ao sistema atual, onde pode existir validação DNSSEC, a diferença consiste no tempo de verificação do certificado TLSA.

- Usabilidade

  Do ponto de vista do utilizador comum, a solução DANE continua a ser extremamente simples pois não prevê nenhuma alteração significativa ao funcionamento dos utilizadores finais.

  Como acontece no sistema atual, do lado do utilizador final existem validações dos certificados digitais. A identificação de um certificado inválido originará um aviso ao utilizador informando a insegurança das comunicações com o domínio em questão.

- Custos de transição

  Visto tratar-se de uma solução que permite uma adaptação ao sistema atual, a solução DANE não comporta custos de transição significativos.

  A obrigatoriedade da implementação DNSSEC e a utilização de *software* capaz de realizar validações DANE constituem os principais custos de transição da solução.

Capítulo 8

Conclusões

8.1 Conclusões

Esta dissertação teve como principal objetivo a definição de uma proposta capaz de substituir ou diminuir o papel das Entidades Certificadoras no sistema de distribuição de chaves criptográficas públicas.

Com essa tarefa em mente, primeiramente foi fundamentada a necessidade da alteração do funcionamento do sistema atual, através da apresentação de um conjunto de situações e fatores capazes de expor os diversos problemas ocasionados pelas CAs no que diz respeito à segurança e autonomia dos utilizadores.

Numa etapa posterior, foi possível demonstrar a possibilidade da implementação de uma estrutura de distribuição de chaves públicas interna através da biblioteca *openSSL*. Por intermédio de uma infraestrutura deste tipo, dirigida por instituições que pela sua identidade possuem um estatuto confiável, conclui-se a viabilidade em quebrar a dependência em CAs externas para se conseguir comunicações seguras.

A análise realizada à U.Porto, relativa ao seu estado de certificação, permitiu comprovar uma dependência, por parte da instituição, a CAs externas na distribuição de chaves públicas para os seus utilizadores. Aceitando essa situação, mas tendo em conta a inclusão da utilização de uma PKI interna, fez-se um conjunto de sugestões de modo a tornar o processo mais eficiente e transparente.

Através do estudo de diversas otimizações/alternativas às CAs concluiu-se que o projeto com maior capacidade de cumprir os objetivos delineados consiste na especificação DANE. Esta solução tem como principais proveitos a possibilidade de adaptação ao sistema atual de CAs, conferindo uma camada adicional de segurança, e a capacidade de romper com a dependência na utilização das CAs através da implementação de certificados auto-assinados. Por outro lado, também se estabeleceu uma orientação para a distribuição direta de chaves públicas utilizando a especificação.

Quanto à proposta apresentada, demonstrou-se a sua viabilidade através de uma prova de conceito e de um conjunto de diretrizes para a implementação da solução.

De uma forma conclusiva, e tendo em consciência a impossibilidade da existência de um sistema completamente seguro a todos os níveis, foi possível constatar a viabilidade da substituição do funcionamento das CAs convencionais tanto num ambiente global, com a implementação do protocolo DANE, como interno, com a utilização de estruturas internas PKI.

8.2 Trabalho Futuro

Os objetivos propostos para um trabalho futuro consistem na análise da implementação de serviços inovadores suportados na tecnologia DANE, tais como o SMIMEA e o OPENPGPKEY.

O projeto SMIMEA[50] consiste na autenticação de correio eletrónico através de s/MIME, onde os certificados ou as suas impressões digitais são armazenados no serviço DNS, seguro através de DNSSEC. O funcionamento tem com base a implementação DANE para TLS, com as devidas alterações para a execução na norma MIME.

O projeto OPENPGPKEY[51] consiste num método DANE para a publicação e localização de chaves PGP no sistema DNS, seguro através de DNSSEC, associada a um endereço de correio eletrónico.

Por fim, propõe-se a implementação da solução DANE num ambiente real, capaz de estabelecer uma prova cabal dos proveitos associados a esta tecnologia, especialmente pelo facto de se basear em DNSSEC, que tem sido alvo de críticas no meios académicos[52, cap. *Design*].

Numa frente alternativa, seria interessante avaliar com mais profundidade a viabilidade da utilização do paradigma da tecnologia *Blockchain* para a construção de um sistema de nomes, incluindo a componente de chaves públicas, a nível mundial. A descentralização da técnica e a sua independência de entidades especificas torna esta possibilidade muito atraente.

Anexo A

Verificações da Implementação *openSSL*

O anexo presente tem como finalidade ilustrar os certificados, os ficheiros de configuração e os processos de instalação dos componentes utilizados na implementação *openSSL* descrita no capítulo 4, bem como os testes que comprovam o seu devido funcionamento.

A.1 Certificados

A.1.1 Certificado da CA Raiz

Os seguintes valores foram introduzidos no momento do pedido da informação por parte do comando referente à criação e assinatura do certificado:

```
Country Name (2 letter code) [XX]: PT
State or Province Name []:  Portugal
Locality Name []:  Porto
Organization Name []:  My Company SA
Organizational Unit Name []:  My Company Root Certificate Authority
Common Name []:  My Company Root CA
Email Address []:admin@mycompany.pt
```

Comando utilizado para a verificação do certificado gerado para a CA raiz:

```
# openssl x509 -noout -text -in ca.cert.pem
```

As figuras A.1 e A.2 representam o cabeçalho e extensões correspondentes à informação do certificado gerada pelo comando anterior.

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number:
            b4:93:4f:56:31:74:c2:09
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=PT, ST=Portugal, L=Porto, O=My Company SA,
        OU=My Company Root Certificate Authority,
        CN=My Company Root CA/emailAddress=admin@mycompany.pt
        Validity
            Not Before: Mar  9 21:09:51 2017 GMT
            Not After : Mar  4 21:09:51 2037 GMT
        Subject: C=PT, ST=Portugal, L=Porto, O=My Company SA,
        OU=My Company Root Certificate Authority,
        CN=My Company Root CA/emailAddress=admin@mycompany.pt
```

Figura A.1: Cabeçalho do certificado da CA raiz

```
X509v3 extensions:
    X509v3 Subject Key Identifier:
        B9:4C:85:66:C0:D3:9B:CB:EA:C5:37:D0:8D:B4:A1:F7:6A:3E:2D:BF
    X509v3 Authority Key Identifier:
        keyid:B9:4C:85:66:C0:D3:9B:CB:EA:C5:37:D0:8D:B4:A1:F7:6A:3E:2D:BF

    X509v3 Basic Constraints: critical
        CA:TRUE
    X509v3 Key Usage: critical
        Digital Signature, Certificate Sign, CRL Sign
```

Figura A.2: Extensões do certificado da CA raiz

A.1.2 Certificado da CA Intermediária

Os seguintes valores foram introduzidos aquando a requisição da informação pelo comando referente à geração do ficheiro de requisição de assinatura:

```
Country Name (2 letter code) [XX]: PT
State or Province Name []:  Portugal
Locality Name []:  Porto
Organization Name []:  My Company SA
Organizational Unit Name []:  My Company Sign Certificate Authority
Common Name []:  My Company Sign CA
Email Address []:admin@mycompany.pt
```

Comando utilizado para a verificação do certificado gerado para a CA intermediária:

```
# openssl x509 -noout -text -in sign_ca.cert.pem
```

As figuras A.3 e A.4 representam o cabeçalho e extensões correspondentes à informação do certificado gerada pelo comando anterior.

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 4096 (0x1000)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=PT, ST=Portugal, L=Porto, O=My Company SA,
        OU=My Company Root Certificate Authority,
        CN=My Company Root CA/emailAddress=admin@mycompany.pt
        Validity
            Not Before: Mar  9 21:33:29 2017 GMT
            Not After : Mar  7 21:33:29 2027 GMT
        Subject: C=PT, ST=Portugal, O=My Company SA,
        OU=My Company Sign Certificate Authority,
        CN=My Company Sign CA/emailAddress=admin@mycompany.pt
```

Figura A.3: Cabeçalho do certificado da CA intermediária

```
X509v3 extensions:
    X509v3 Subject Key Identifier:
        34:5B:C7:C9:8C:87:41:A7:04:00:04:8F:37:68:CB:DD:41:3A:EA:2C
    X509v3 Authority Key Identifier:
        keyid:B9:4C:85:66:C0:D3:9B:CB:EA:C5:37:D0:8D:B4:A1:F7:6A:3E:2D:BF

    X509v3 Basic Constraints: critical
        CA:TRUE, pathlen:0
    X509v3 Key Usage: critical
        Digital Signature, Certificate Sign, CRL Sign
```

Figura A.4: Extensões do certificado da CA intermediária

A.1.3 Certificado da Página *www.mycompany.pt*

Os seguintes valores foram introduzidos aquando a requisição da informação pelo comando referente à geração do ficheiro de requisição de assinatura:

```
Country Name (2 letter code) [XX]: PT
State or Province Name []:  Portugal
Locality Name []:  Porto
Organization Name []:  My Company SA
Organizational Unit Name []:  My Company SA Web Services
Common Name []:  www.mycompany.pt
Email Address []:admin@mycompany.pt
```

Comando utilizado para a verificação do certificado gerado para a página *www.mycompany.pt*.

```
# openssl x509 -text -noout -in www.mycompany.pt.cert.pem
```

As figuras A.5 e A.6 representam o cabeçalho e extensões correspondentes à informação do certificado gerada pelo comando anterior.

Na figura A.6, referente às extensões do certificado, é possível verificar os apontadores para os serviços de revogação de certificados da estrutura, mais concretamente a lista de certificados revogados e o servidor OCSP.

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 4107 (0x100b)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=PT, ST=Portugal, O=My Company SA,
        OU=My Company Sign Certificate Authority,
        CN=My Company Sign CA/emailAddress=admin@mycompany.pt
        Validity
            Not Before: Mar 17 02:25:43 2017 GMT
            Not After : Mar 17 02:25:43 2018 GMT
        Subject: C=PT, ST=Portugal, L=Porto, O=My Company SA,
        OU=My Company Web Services,
        CN=www.mycompany.pt/emailAddress=admin@mycompany.pt
```

Figura A.5: Cabeçalho do certificado da *www.mycompany.pt*

```
X509v3 extensions:
    X509v3 Basic Constraints:
        CA:FALSE
    X509v3 Subject Key Identifier:
        0C:8C:71:90:D7:D0:0A:55:55:04:34:90:6B:D6:E4:42:8D:49:D6:EE
    X509v3 Authority Key Identifier:
        keyid:34:5B:C7:C9:8C:87:41:A7:04:00:04:8F:37:68:CB:DD:41:3A:EA:2C

    X509v3 Key Usage: critical
        Digital Signature, Key Encipherment
    X509v3 Extended Key Usage:
        TLS Web Server Authentication
    X509v3 CRL Distribution Points:

        Full Name:
          URI:http://crl.mycompany.pt/sign-ca.crl.pem

    Authority Information Access:
        OCSP - URI:http://ocsp.mycompany.pt
```

Figura A.6: Extensões do certificado da página *www.mycompany.pt*

A.1.4 Certificados do Utilizador *Frank*

Os seguintes valores foram introduzidos aquando a requisição da informação pelo comando referente à geração do ficheiro de requisição de assinatura:

```
Country Name (2 letter code) [XX]: PT
State or Province Name []:  Portugal
Locality Name []:  Porto
Organization Name []:  My Company SA
Organizational Unit Name []:
Common Name []:  frank@mycompany.pt
Email Address []:  frank@mycompany.pt
```

Comando utilizado para a verificação do certificado gerado para o utilizador *Frank*.

```
# openssl x509 -text -noout -in frank.mycompany.cert.pem
```

As figuras A.7 e A.8 representam o cabeçalho e extensões correspondentes à informação do certificado gerada pelo comando anterior.

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 4104 (0x1008)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=PT, ST=Portugal, O=My Company SA,
        OU=My Company Sign Certificate Authority,
        CN=My Company Sign CA/emailAddress=admin@mycompany.pt
        Validity
            Not Before: Mar 16 04:56:22 2017 GMT
            Not After : Mar 16 04:56:22 2018 GMT
        Subject: C=PT, ST=Portugal, L=Porto, O=My Company SA,
        CN=frank@mycompany.pt/emailAddress=frank@mycompany.pt
```

Figura A.7: Cabeçalho do certificado do utilizador *Frank*

Na figura A.8, referente às extensões do certificado, é possível verificar os apontadores para os serviços de revogação de certificados da estrutura, mais concretamente a lista de certificados revogados e o servidor OCSP.

```
X509v3 extensions:
    X509v3 Basic Constraints:
        CA:FALSE
    X509v3 Subject Key Identifier:
        F2:F5:88:5A:B3:57:AA:8B:42:E8:01:8E:54:41:9F:CE:0E:C8:5F:5E
    X509v3 Authority Key Identifier:
        keyid:34:5B:C7:C9:8C:87:41:A7:04:00:04:8F:37:68:CB:DD:41:3A:EA:2C

    X509v3 Key Usage: critical
        Digital Signature, Non Repudiation, Key Encipherment
    X509v3 Extended Key Usage:
        TLS Web Client Authentication, E-mail Protection
    X509v3 CRL Distribution Points:

        Full Name:
          URI:http://crl.mycompany.pt/sign-ca.crl.pem

    Authority Information Access:
        OCSP - URI:http://ocsp.mycompany.pt
```

Figura A.8: Extensões do certificado do utilizador *Frank*

A.1.5 Certificado do servidor OCSP

Os seguintes valores foram introduzidos aquando a requisição da informação pelo comando referente à geração do ficheiro de requisição de assinatura:

```
Country Name (2 letter code) [XX]: PT
State or Province Name []:  Portugal
Locality Name []:  Porto
Organization Name []:  My Company SA
Organizational Unit Name []:  My Company Certificate Authority
Common Name []:  ocsp.mycompany.pt
Email Address []:admin@mycompany.pt
```

Comando utilizado para a verificação do certificado gerado para o servidor OCSP.

```
# openssl x509 -text -noout -in ocsp.mycompany.pt.cert
```

As figuras A.9 e A.10 representam o cabeçalho e extensões correspondentes à informação do certificado gerada pelo comando anterior.

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 4109 (0x100d)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=PT, ST=Portugal, O=My Company SA,
        OU=My Company Sign Certificate Authority,
        CN=My Company Sign CA/emailAddress=admin@mycompany.pt
        Validity
            Not Before: Mar 17 04:11:02 2017 GMT
            Not After : Mar 27 04:11:02 2018 GMT
        Subject: C=PT, ST=Portugal, L=Porto, O=My Company SA,
        OU=My Company Certificate Authority,
        CN=ocsp.mycompany.pt/emailAddress=admin@mycompany.pt
```

Figura A.9: Cabeçalho do certificado do servidor OCSP

```
X509v3 extensions:
    X509v3 Basic Constraints:
        CA:FALSE
    X509v3 Subject Key Identifier:
        3E:EA:A5:FF:05:2E:17:F2:A2:12:3D:F0:89:E7:80:D8:F4:B3:51:D6
    X509v3 Authority Key Identifier:
        keyid:34:5B:C7:C9:8C:87:41:A7:04:00:04:8F:37:68:CB:DD:41:3A:EA:2C

    X509v3 Key Usage: critical
        Digital Signature
    X509v3 Extended Key Usage: critical
        OCSP Signing
```

Figura A.10: Extensões do certificado do servidor OCSP

A.2 Ficheiros de configuração

A.2.1 CA raiz

Principais secções do ficheiro:

- Política

```
countryName = match
stateOrProvinceName = match
organizationName = match
organizationalUnitName = optional
commonName = supplied
emailAddress = optional
```

A CA raiz apenas assina certificados em que os campos de *countryName*, *stateOrProvinceName* e *organizationName* coincidam com os valores dos campos do certificado raiz. O *commonName* é registado com o nome providenciado na solicitação do certificado.

  - *countryName*: Abreviatura de 2 letras ISO para o país
  - *stateOrProvinceName*: Estado ou província
  - *organizationName*: Nome da organização
  - *organizationalUnitName*: Secção da organização
  - *commonName*: Nome da entidade alvo do processo de certificação
  - *emailAddress*: Endereço de correio eletrónico

- Extensões

Extensão utilizada para a criação do certificado da CA raiz:

```
[ ca_ext ]
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always
basicConstraints = critical, CA:true
keyUsage = critical, digitalSignature, cRLSign, keyCertSign
```

Extensão utilizada para a criação do certificado da CA intermediária:

```
[ sign_ca_ext ]
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always
basicConstraints = critical, CA:true, pathlen:0
keyUsage = critical, digitalSignature, cRLSign, keyCertSign
```

Explicitação dos campos utilizados nas extensões:

- *subjectKeyIdentifier*: Valor: *hash*. Consiste num valor derivado da chave pública do visado.
- *authorityKeyIdentifier*: Valores possíveis: *keyid* e *issuer*. Se *keyid*, é realizada uma tentativa de cópia do *subject key identifier* do certificado da entidade emissora. Se o valor *always* estiver presente, é apresentado um erro se esta opção falhar. Se *issuer*, o emissor e número de serie são copiados do certificado da entidade emissora.
- *basicConstraints*: Indicador para determinar se o certificado é um certificado de uma CA. A indicação é apresentada através do parâmetro CA (*true* ou *false*). O parâmetro *pathlen* indica o numero máximo de CAs que podem aparecer seguidas desta, num caminho de certificados.
- *keyUsage*: Lista das possíveis utilizações permitidas. Para ambos os casos, para os certificados emitidos são permitidos ações de assinatura digital, assinatura de listas de revogação de certificados e assinatura de certificados.

A.2.2 CA intermediária

Principais secções do ficheiro:

- Política

```
countryName = match
stateOrProvinceName = match
organizationName = match
organizationalUnitName = optional
commonName = supplied
emailAddress = optional
```

Tal como a CA raiz, a CA intermediária apenas assina certificados em que os campos de *countryName*, *stateOrProvinceName* e *organizationName* coincidam com os valores dos campos do seu próprio certificado. O *commonName* é registado com o nome providenciado na solicitação do certificado.

- Extensões

  Extensão utilizada para a criação de certificados para utilizadores:

```
[ user_cert_ext ]
basicConstraints = CA:FALSE
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always
keyUsage = critical, nonRepudiation, digitalSignature, keyEncipherment
extendedKeyUsage = clientAuth, emailProtection
crlDistributionPoints = URI:http://crl.mycompany.pt/sign-ca.crl.pem
authorityInfoAccess = OCSP;URI:http://ocsp.mycompany.pt
```

  Extensão utilizada para a criação de certificados para servidores:

```
[ server_cert_ext ]
basicConstraints = CA:FALSE
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always
keyUsage = critical, digitalSignature, keyEncipherment
extendedKeyUsage = serverAuth
crlDistributionPoints = URI:http://crl.mycompany.pt/sign-ca.crl.pem
authorityInfoAccess = OCSP;URI:http://ocsp.mycompany.pt
```

  Extensão utilizada para identificação da lista de certificados revogados:

```
[ crl_ext ]
authorityKeyIdentifier=keyid:always
```

  Extensão utilizada para identificação do servidor OCSP:

```
[ ocsp_ext ]
basicConstraints = CA:FALSE
subjectKeyIdentifier = hash
authorityKeyIdentifier = keyid:always
keyUsage = critical, digitalSignature
extendedKeyUsage = critical, OCSPSigning
```

Explicitação dos campos utilizados nas extensões:

- *crlDistributionPoints*: Indicador do URL de acesso à lista de certificados revogados
- *authorityInfoAccess*: Indicador do URL de acesso ao servidor OCSP
- *keyUsage*: Lista das possíveis utilizações da chave pública do certificado. No caso da extensão de clientes, são permitidos ações de não repúdio, assinatura digital e encriptação. No caso da extensão para servidores são permitidas ações de assinatura digital e encriptação. No caso da extensão para servidores OCSP são permitidas ações de assinatura digital.
- *extendedKeyUsage*: Lista alargada de possíveis utilizações da chave pública do certificado. No caso da extensão para clientes são permitidas ações de autenticação de clientes e proteção de correio eletrónico. No caso da extensão para servidores são permitidas ações de autenticação de servidores. No caso da extensão para servidores OCSP são permitidas ações de assinatura de respostas OCSP.

A.3 Processos de Instalação dos Componentes

A.3.1 Instalação do Servidor *Web*

O servidor *web* foi instalado através do *software apache2* na máquina *Debian 2*.

Depois da entrega por parte da CA intermediária, o certificado referente ao servidor *web* foi instalado no servidor de modo a providenciar acessos via HTTPS à página *www.mycompany.pt*. Para ativar o protocolo HTTPS no servidor foi necessário: ativar o modo SSL, importar o certificado e respetiva chave para a diretoria do servidor e editar o ficheiro de configurações do domínio de modo a comunicar através da porta 443.

A.3.2 Instalação do Servidor de Correio Eletrónico

O servidor de correio eletrónico foi instalado através do *software postfix* na máquina *Debian 2*.

Nas máquinas *Debian 3* e *Debian 4* foi instalado o cliente de correio eletrónico *Icedove* de modo a possibilitar os acessos dos utilizadores às contas configuradas no servidor.

Depois da entrega por parte da CA intermediária, os certificados referentes aos utilizadores foram instalados nos clientes de correio eletrónico dos respetivos utilizadores de modo a providenciar trocas de mensagens de correio eletrónico encriptadas com as chaves dos utilizadores correspondentes.

A.3.3 Instalação da Lista de Certificados Revogados no Servidor *Web*

De modo a instalar a lista de certificados revogados, foi utilizado o *software apache2* para criar uma página *web* alojada no servidor *web* da máquina *Debian 2* com a lista de certificados revogados disponível para descarregamento.

A.3.4 Instalação do Servidor OCSP

Os acessos ao servidor OCSP são realizados através de HTTP (porta 80). Visto que o servidor OCSP foi configurado de modo a escutar a porta 2560, foi necessário realizar um redirecionamento do tráfego da porta 80 para a porta 2560, para pedidos ao servidor OCSP.

Linhas para o redirecionamento do ficheiro de configuração do *software apache2*:

```
RewriteEngine on
RewriteRule ^/(.)  http://ocsp.mycompany.pt:2560/ [P]
```

Foi utilizado o comando seguinte para a ativação do servidor OCSP:

```
# openssl ocsp -index index.txt -port 2560 -rsigner
ocsp.mycompany.pt.cert.pem -rkey ocsp.mycompany.key.pem -CA
sign-ca.cert.pem
```

No comando anterior, é utilizado o ficheiro que contém a informação dos certificados, `index.txt`, o certificado assinante (`ocsp.mycompany.pt.cert.pem`), a chave correspondente (`ocsp.mycompany.key.pem`) e o certificado da CA correspondente à informação de revogação `/sign-ca.cert.pem`).

A razão da necessidade da inclusão do certificado e da chave correspondente reside no facto de que todas as repostas serem assinadas pelo servidor OCSP.

A.4 Testes

A.4.1 Acesso à Página *www.mycompany.pt*

De modo a aceder à página *www.mycompany.pt* através do protocolo HTTPS, foi necessário estabelecer uma relação de confiança com a CA do sistema. Essa relação foi realizada através da instalação, no navegador *web*, dos certificados referentes à CA raiz e à CA intermediária, tal como ilustra a figura A.11.

A figura A.12 representa o essencial dos certificados instalados no serviço de gestão de certificados referente ao navegador *web* da máquina *Debian 3*.

A figura A.13 representa o acesso na máquina *Debian 3* à página *www.mycompany.pt* através do protocolo HTTPS. Nela é possível verificar, na barra de endereços do navegador, o protocolo utilizado.

Depois de realizado um acesso à página *www.mycompany.pt* através do navegador *web* procedeu-se à visualização do certificado descarregado. O certificado está representado na figura A.14, que pela sua informação comprova o sucesso da sua implementação.

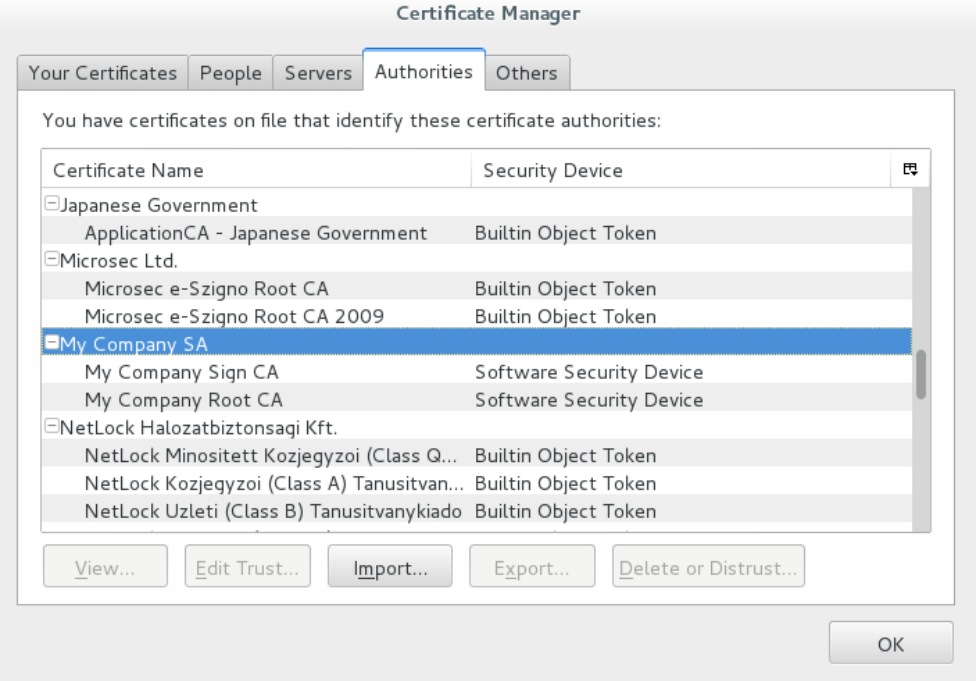


Figura A.11: Serviço de gestão de certificados *Firefox*

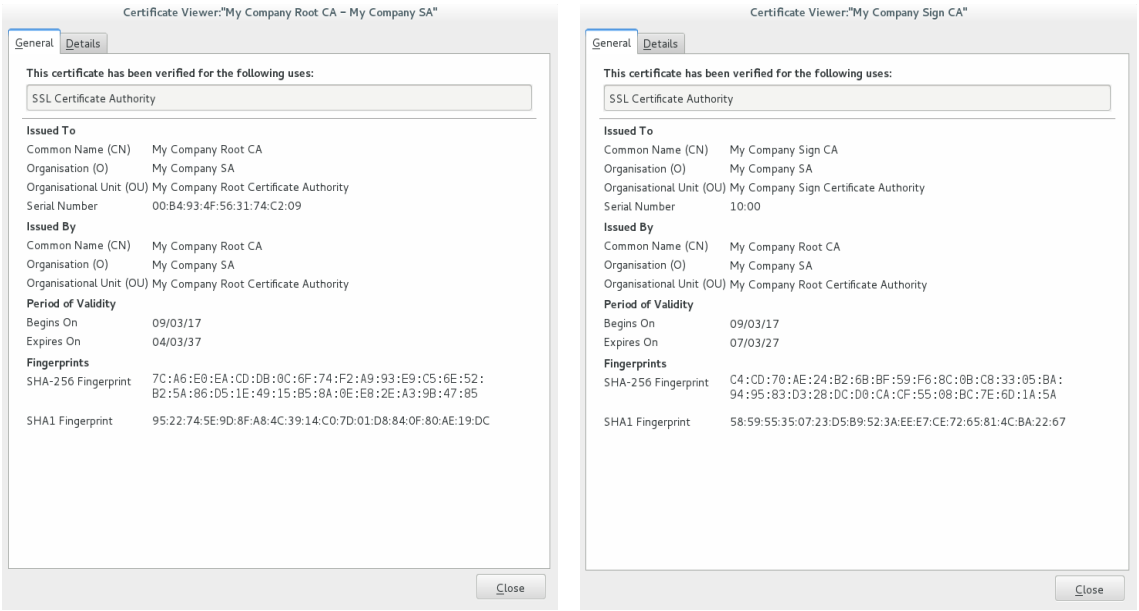


Figura A.12: Certificados da CA raiz e CA intermediária no serviço de gestão de certificados *Firefox*

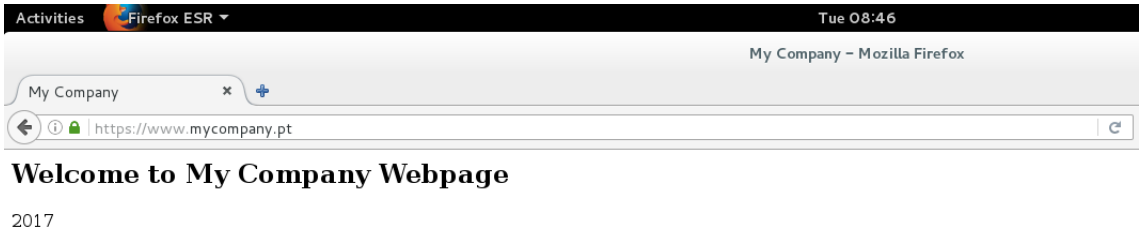


Figura A.13: Acesso HTTPS página *www.mycompany.pt*

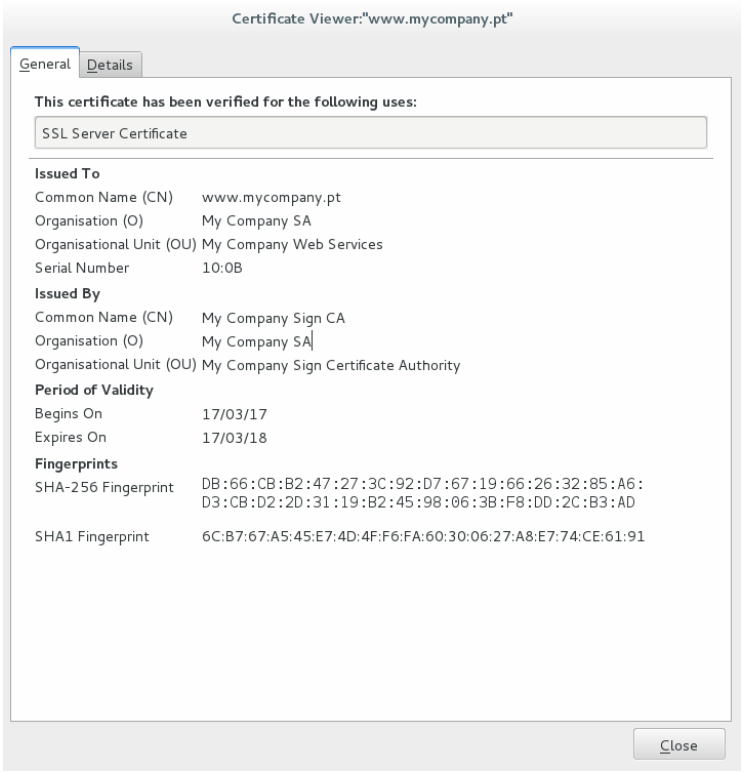


Figura A.14: Certificado descarregado da página *www.mycompany.pt*

A.4.2 Trocas de Correio Eletrónico

De modo a trocar mensagens de correio eletrónico de forma segura, foi necessário estabelecer uma relação de confiança com a CA do sistema. Para isso, foram instalados no cliente de correio eletrónico *Icedove*, os certificados referentes à CA raiz e à CA intermediária, tal como ilustra a figura A.15.

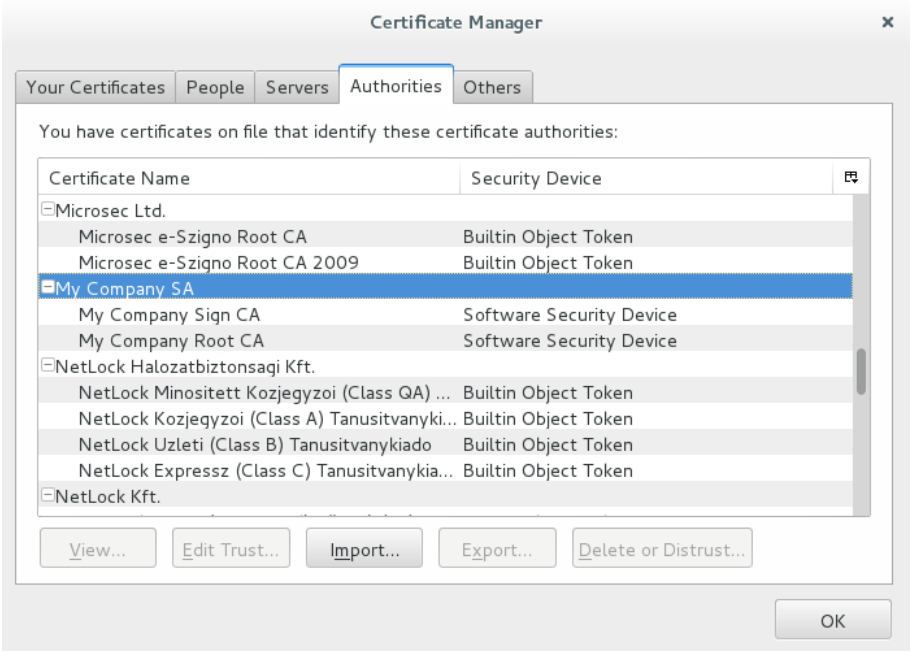


Figura A.15: Serviço de gestão de certificados *Icedove*

De seguida, na conta do utilizador *Sam*, procedeu-se à instalação do certificado associado a esse utilizador no serviço de gestão de certificados, de modo a poder realizar procedimentos de encriptação com a chave pública a si associada. O resultado da instalação encontra-se representado na figura A.16.

A figura A.17 representa os detalhes do certificado do utilizador final *Sam* no serviço de gestão de certificados referente ao cliente de correio eletrónico *Icedove*.

O certificado previamente instalado é utilizado para a encriptação e desencriptação de mensagens do utilizador Sam.

Na figura A.18 encontra-se representado um envio de uma mensagem de correio eletrónico desde a máquina *Debian 4* (utilizador *Frank*) para a máquina *Debian 3* (utilizador *Sam*). A mensagem foi encriptada e assinada digitalmente.

A receção do correio eletrónico enviado pelo utilizador *Frank* encontra-se ilustrada na figura A.19. Nela é possível verificar que a mensagem providenciada pelo cliente de correio eletrónico informando que a mensagem recebida encontra-se assinada e encriptada.

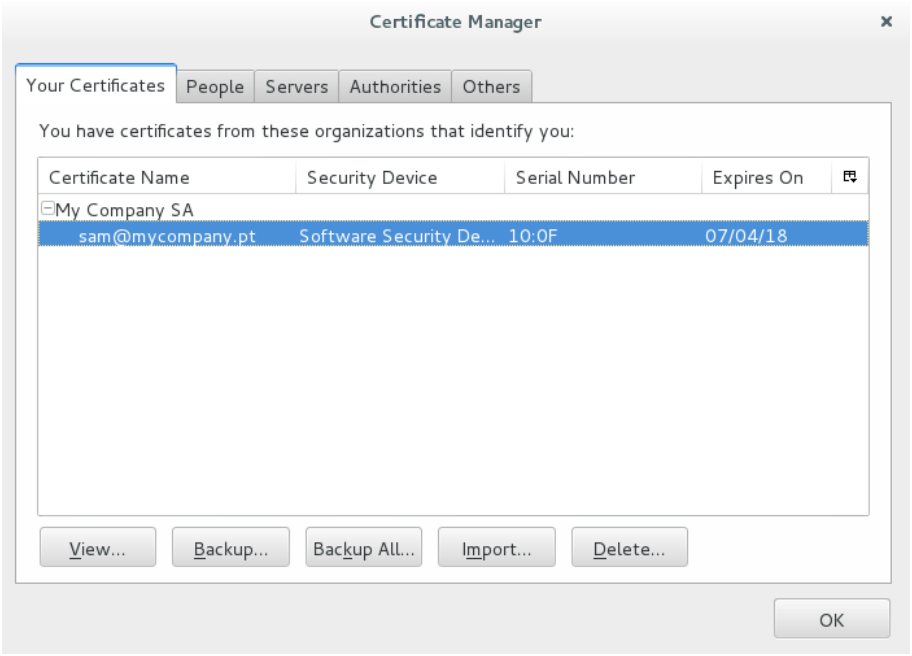


Figura A.16: Instalação do certificado do utilizador *Sam*

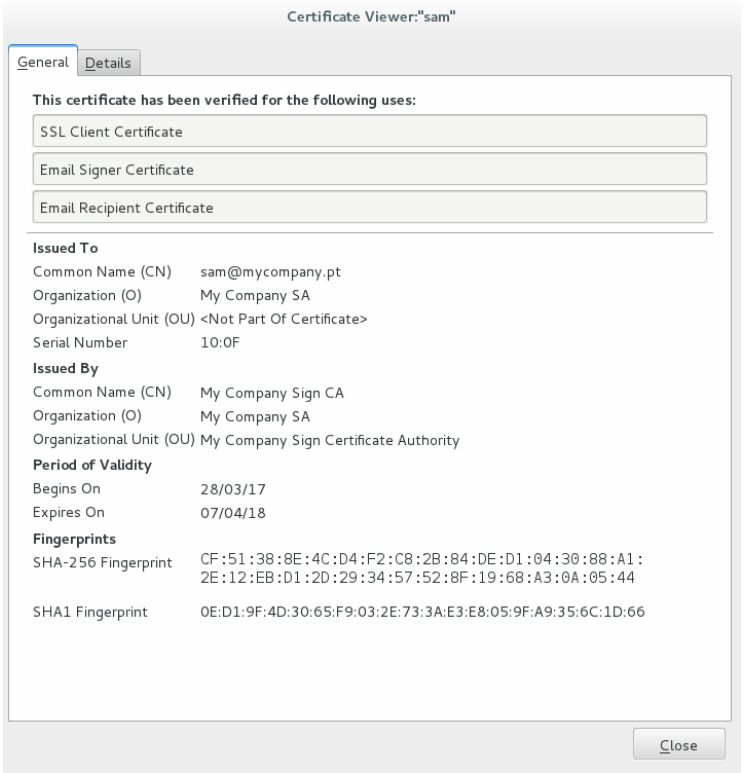


Figura A.17: Certificado do utilizador *Sam* no serviço de gestão de certificados *Icedove*

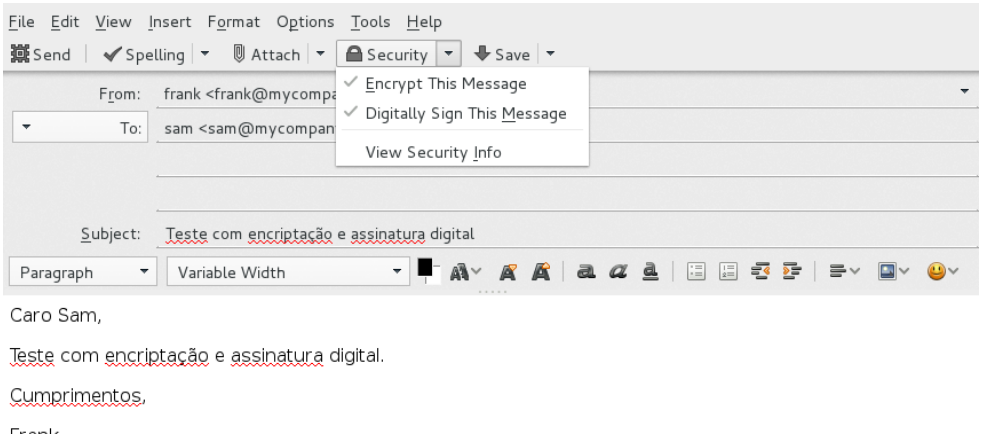


Figura A.18: Envio correio eletrónico

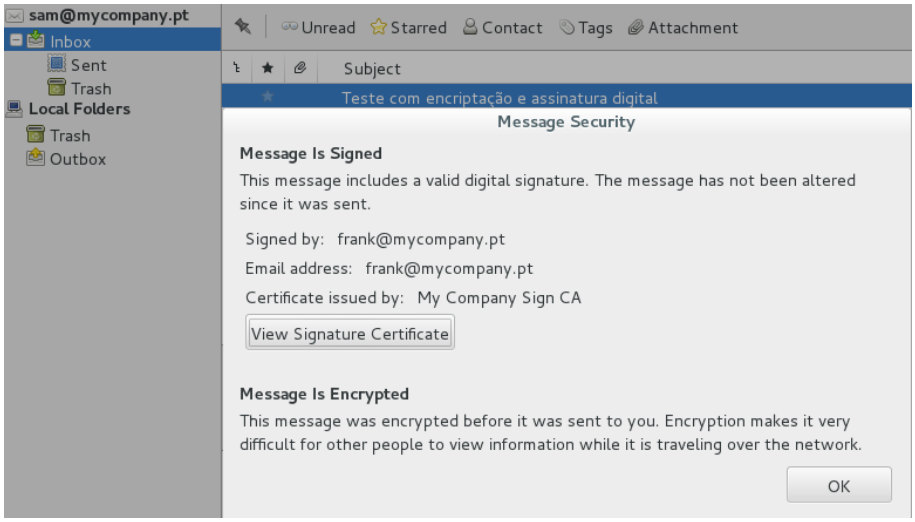


Figura A.19: Receção correio eletrónico

A.4.3 Acesso à Lista de Certificados Revogados no Servidor *Web*

De modo a instalar a lista de certificados revogados, foi utilizado o *software apache2* para criar uma página *web* alojada no servidor *web* da máquina *Debian 2*.

Tal como indica a figura A.20 para aceder à lista de certificados revogados, foi utilizado o endereço: `http://crl.mycompany.pt/sign-ca.crl.pem`.

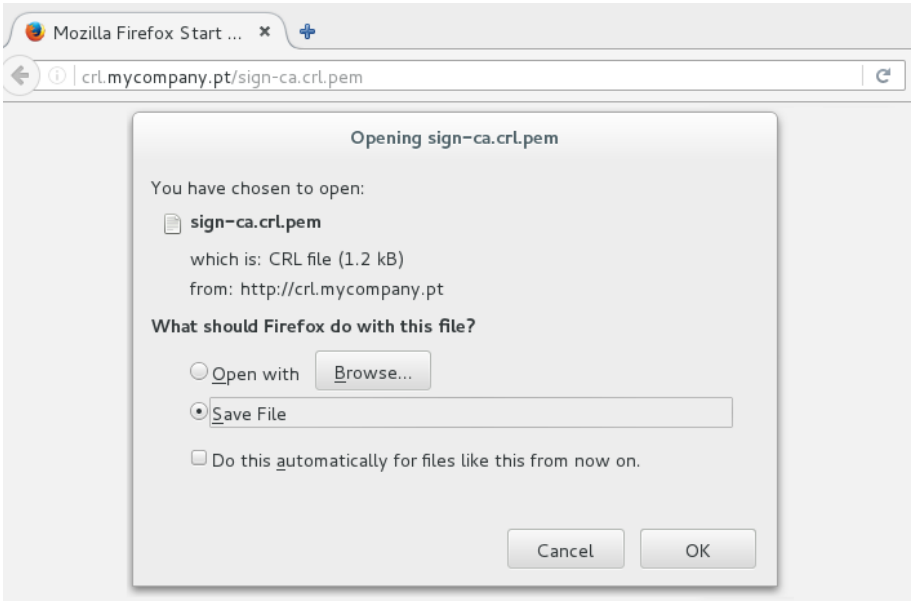


Figura A.20: Lista de certificados revogados

A.4.4 Acesso ao Servidor OCSP através de Comandos

A seguinte prova consiste num teste ao servidor OCSP com um pedido de estado de um certificado revogado de modo a validar a reposta por parte do servidor.

Para ativar manualmente o servidor OCSP e escutar pedidos na porta 2560 foi executado o seguinte comando:

```
# openssl ocsp -CAfile root-sign.cert.pem -url
http://ocsp.mycompany.pt:2560 -resp_text -issuer sign-ca.cert.pem -cert
www.test-mycompany.pt.cert.pem
```

No comando é utilizado o certificado `root-sign.cert.pem` que consiste numa concatenação dos certificados da CA raiz e da CA intermediária. Este certificado serve para verificar a assinatura da resposta OCSP. No comando também é indicado o caminho para o servidor OCSP, a opção `-resp_text` de modo a apresentar o output, o certificado do emissor, `sign-ca.cert.pem` e o certificado ao qual se pretende conhecer o estado, `www.test-mycompany.pt.cert.pem`.

A saída do comando anterior consiste na seguinte informação:

```
Response verify OK
newcerts/www.test-mycompany.pt.cert.pem:  revoked
This Update:  Apr 5 20:41:58 2017 GMT
Revocation Time:  Mar 17 03:04:46 2017 GMT
```

De modo a realizar um teste ao servidor OCSP com um pedido de estado de um certificado válido, foi utilizado o seguinte comando:

```
# openssl ocsp -CAfile root-sign.cert.pem -url http://ocsp.mycompany.pt:2560
-resp_text -issuer sign-ca.cert.pem -cert www.mycompany.pt.cert.pem
```

A saída do comando anterior consiste na seguinte informação:

```
Response verify OK
newcerts/www.mycompany.pt.cert.pem:  good
This Update:  Apr 5 20:51:28 2017 GMT
```

A.4.5 Acesso ao Servidor OCSP através do Navegador *Web*

A seguinte prova consiste num teste ao servidor OCSP através de um acesso na máquina *Debian 3* à página *www.mycompany.pt* via um navegador *web*.

O navegador *web* realiza um pedido ao servidor OCSP de modo a validar o estado do certificado apresentado pela página *www.mycompany.pt*. Nas figuras A.21 e A.22 é possível identificar, respetivamente, a consulta e a resposta OCSP dos dados referentes ao certificado da página *www.mycompany.pt*, com o número de série 4107 (*0x100b*), através do *software Wireshark*.

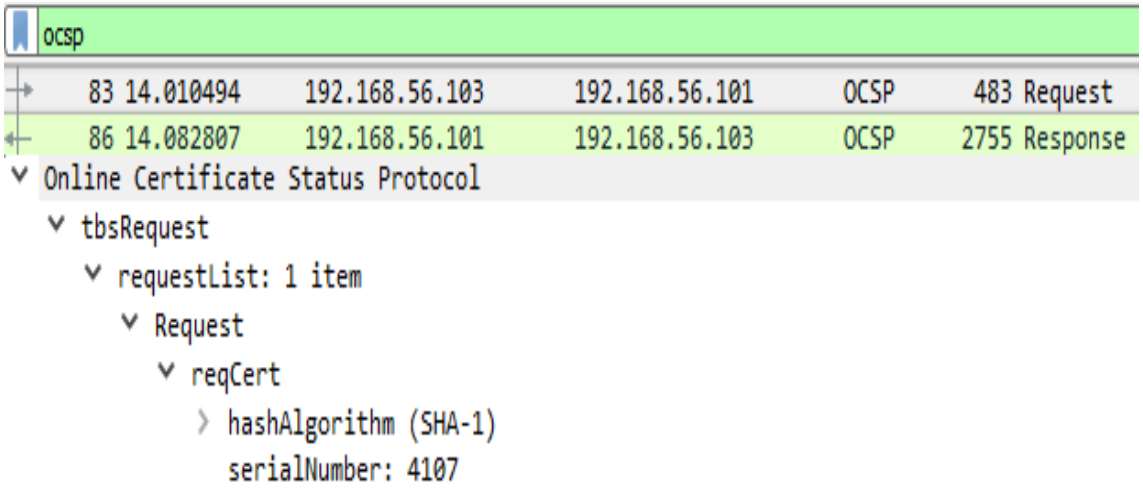


Figura A.21: Requisição OCSP - *Wireshark*

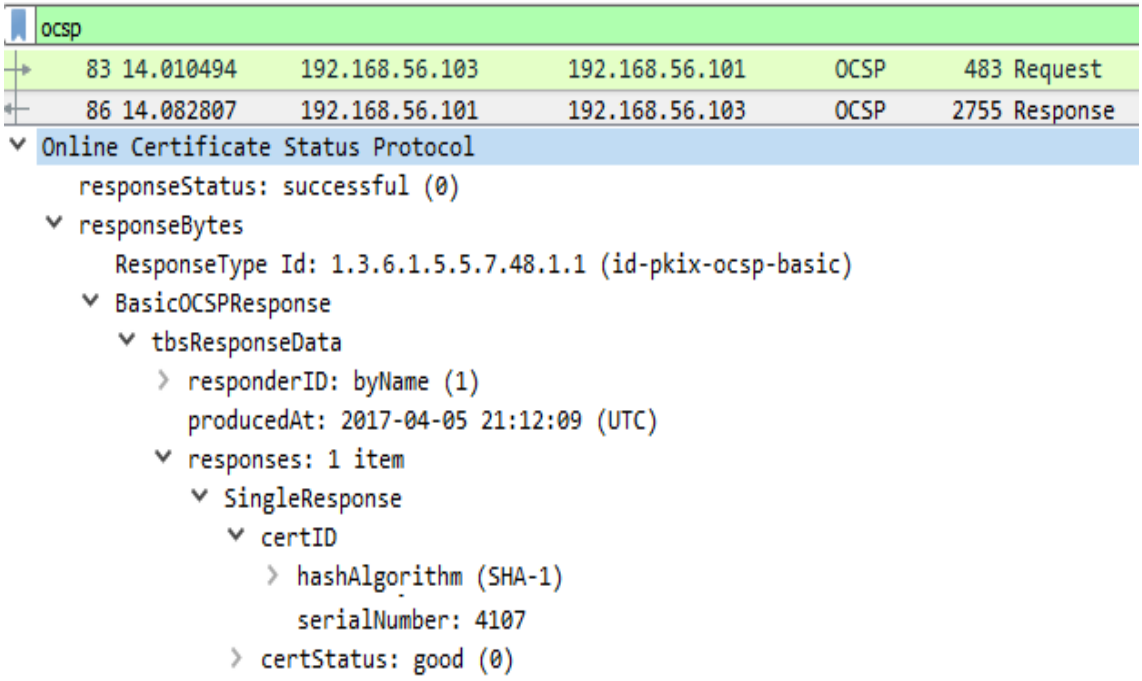


Figura A.22: Resposta OCSP - *Wireshark*

Anexo B

Verificações da Implementação DANE

O anexo presente tem como finalidade ilustrar os diferentes componentes associados à prova de conceito descrita no capítulo 7.

B.1 Servidores DNS

Configuração de um servidor DNS autoritário e recursivo nas máquinas *Debian 5* e *Debian 4*, respetivamente.

Debian 5

Para a máquina *Debian 5*, correspondente ao servidor DNS autoritário, foram realizadas as seguintes configurações:

1. Ficheiro `named.conf.local`

   Neste ficheiro foi indicada a zona à qual o servidor autoritário é responsável por responder a consultas dos utilizadores da rede. Como indicado na estrutura seguinte, o nome da zona corresponde a `exemplo.fms`, o servidor consiste no detentor da informação e o ficheiro com os registos DNS situa-se no diretório `/etc/bind/db.exemplo.fms`.

   ```
   zone "exemplo.fms"{
   type master;
   file "/etc/bind/db.exemplo.fms;
   };
   ```

2. Ficheiro `db.exemplo.fms`

   O ficheiro `db.exemplo.fms` foi criado para armazenar a informação acerca dos registos DNS da zona `exemplo.fms`. Nesta zona estão definidas as indicações do *nameserver* `exemplo.fms`, do endereço de IP do domínio `exemplo.fms` e do endereço de IP para o domínio *www.exemplo.fms*, juntamente com informações gerais do ficheiro.

```
; BIND data file for local loopback interface
;
$TTL 604800
@ IN SOA exemplo.fms.  root.exemplo.fms.  (
28 ; Serial
604800 ; Refresh
86400 ; Retry
2419200 ; Expire
604800 ) ; Negative Cache TTL
;
@ IN NS exemplo.fms.
@ IN A 192.168.56.105
www IN A 192.168.56.106
```

Debian 4

Para a máquina *Debian 4*, que corresponde ao servidor DNS recursivo, foram realizadas as seguintes configurações:

1. Ficheiro `named.conf.options`

   Com o intuito de direcionar todas as consultas para o servidor autoritário da rede, foi indicado, neste ficheiro, o servidor DNS autoritário:

   ```
   forwarders
   192.168.56.105;
   ;
   ```

B.2 Implementação do DNSSEC

Para realizar a geração das chaves e dos registos DNSSEC de uma forma automática procedeu-se à utilização do pacote DNSSEC-TOOLS, mais precisamente da ferramenta *zonesigner*. Esta ferramenta, como o próprio nome indica, assina a zona passada por parâmetro, gerando todos os registos necessários para a implementação DNSSEC.

Debian 5

1. *zonesigner*

   Devido ao fato de que a zona `exemplo.fms` estar definida na máquina *Debian 5*, no diretório correspondente ao serviço DNS desta máquina foi executado o seguinte comando:

   ```
   # zonesigner -genkeys -zone exemplo.fms db.exemplo.fms
   ```

Como se trata da primeira ocorrência da ferramenta na zona, o comando é invocado com o parâmetro de geração de chaves `genkeys` para a zona `exemplo.fms` no `ficheiro db.exemplo.fms`. Depois do momento da geração das chaves, qualquer alteração ao ficheiro `db.exemplo.fms`, deve ser seguida do comando *zonesigner* anterior sem o parâmetro de geração de chaves (*genkeys*).

Na figura B.1 está representado o diretório associado ao serviço DNS da máquina *Debian 5*, depois da execução do comando *zonesigner*, onde é possível verificar a geração com sucesso das várias chaves. O ficheiro `db.exemplo.fms.signed` também foi gerado aquando a execução do comando e contém os registos da zona juntamente com os registos referentes à tecnologia DNSSEC.

```
root@debian5:/etc/bind# ls
bind.keys                    Kexemplo.fms.+008+38976.private
db.0                         Kexemplo.fms.+008+40462.key
db.127                       Kexemplo.fms.+008+40462.private
db.255                       Kexemplo.fms.+008+44496.key
db.empty                     Kexemplo.fms.+008+44496.private
db.exemplo.fms               named.conf
db.exemplo.fms.signed        named.conf.default-zones
db.local                     named.conf.local
db.root                      named.conf.options
dsset-exemplo.fms.           rndc.key
exemplo.fms.krf              zones.rfc1918
Kexemplo.fms.+008+38976.key
```

Figura B.1: Diretório serviço DNS - *Debian 5*

2. Alteração do ficheiro `named.conf.local`

   Na máquina *Debian 5*, o ficheiro `named.conf.local`, que contém a indicação da zona `exemplo.fms` e o apontador para o ficheiro onde estão armazenados os registos da zona, foi alterado para apontar para o ficheiro gerado pelo comando *zonesigner*, `db.exemplo.fms.signed`.

```
zone "exemplo.fms"{
type master;
file "/etc/bind/db.exemplo.fms.signed;
};
```

Debian 4

1. Indicação no servidor recursivo da chave do servidor autoritário

   Com o objetivo de possibilitar a autenticação da informação proveniente do servidor autoritário, o servidor recursivo deve deter em sua posse a chave pública KSK do servidor autoritário. Neste caso, este processo consiste na obtenção da KSK do servidor autoritário armazenada no ficheiro `db.exemplo.fms.signed` da máquina *Debian 5* seguida do

seu armazenamento na máquina *Debian 4*, no ficheiro `named.conf.options`. O seguinte bloco representa a informação referente à KSK do servidor autoritário no servidor recursivo.

```
trusted-keys{
exemplo.fms.  257 3 8 "AwEAAbGDGmURzPeTRLJB6LjS2rnMZ5+PGGSVwPeZzSdA
2D3cphRRsA77cKqlnDsW0ryc5TVxdz2wCl5qTktuBkhNcY4tvKWYx9KEZXA7qb0f7qK
XhPTq/lhFeMztDgWzYMJOfWdfBGBWomn3z25quqHYwT9Z2Fn61Ds/fQnFShYkn7Ca00
56 ER6tlAs3jfysdEqFY02jN8D0VqTzB9IEN0qS0b87wF00++Y95nek497QU6pIfCXK
WVhXKIw0 9iL74mEtRnuEFF3McQ8Uq7WtIkNH6mn70xP/aaB8MKysqv4K9IrpqF+PkI
eKM/n0BhO6D8/Q gtLlW8zSQpFQ+kRlXc5O+d0=";
};
```

A *tag* 257 indica que se trata da chave utilizada para assinar outras chaves (KSK).

2. Verificação DNSSEC

   A verificação DNSSEC foi realizada na máquina *Debian 4* através do comando `dig` seguido do domínio ao qual se pretende traduzir o endereço de IP.

   Na figura B.2, é possível verificar o comando realizado, a secção de resposta que indica o endereço de IP correto, o servidor DNS, etc. A informação mais importante, no que diz respeito à validade da implementação DNSSEC, trata-se da indicação das *flags* no *header* da resposta. A existência da *flag ad* indica que o domínio encontra-se assinado via DNSSEC e foi validado com sucesso.

```
root@debian4:/etc/bind# dig www.exemplo.fms

; <<>> DiG 9.9.5-9+deb8u11-Debian <<>> www.exemplo.fms
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 51684
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;www.exemplo.fms.               IN      A

;; ANSWER SECTION:
www.exemplo.fms.        604800  IN      A       192.168.56.106

;; AUTHORITY SECTION:
exemplo.fms.            601201  IN      NS      exemplo.fms.

;; ADDITIONAL SECTION:
exemplo.fms.            604800  IN      A       192.168.56.105

;; Query time: 5 msec
;; SERVER: 192.168.56.104#53(192.168.56.104)
;; WHEN: Tue Jun 06 04:53:24 BST 2017
;; MSG SIZE  rcvd: 90
```

Figura B.2: Resposta à consulta do domínio - *Debian 4*

B.3 Servidor HTTP

Para configurar o servidor *web* foi instalada a ferramenta *apache2* na máquina *Debian 6*. O primeiro passo para a sua configuração consiste na criação dos ficheiros fonte (*html, css*, etc). Neste caso foi criado um ficheiro *html* com uma estrutura visual simples.

O passo seguinte consistiu na criação do ficheiro `exemplo.conf` no diretório `/etc/apache2/sites-available` que contém as principais definições da página, tais como: a porta 80 para o protocolo HTTP, o nome do servidor, o diretório onde se encontram localizados os ficheiros fonte da página e a indicação para os registos de *log*.

Para a ativação do protocolo TLS para a página foi necessário proceder à criação de um certificado para o domínio em questão. Para isso foi utilizada a estrutura criada no capitulo 4 do documento de modo a requisitar a assinatura do certificado por parte de uma CA. A figura B.3 representa o certificado assinado pela CA intermediária *My Company Sign* para a página *www.exemplo.fms*.

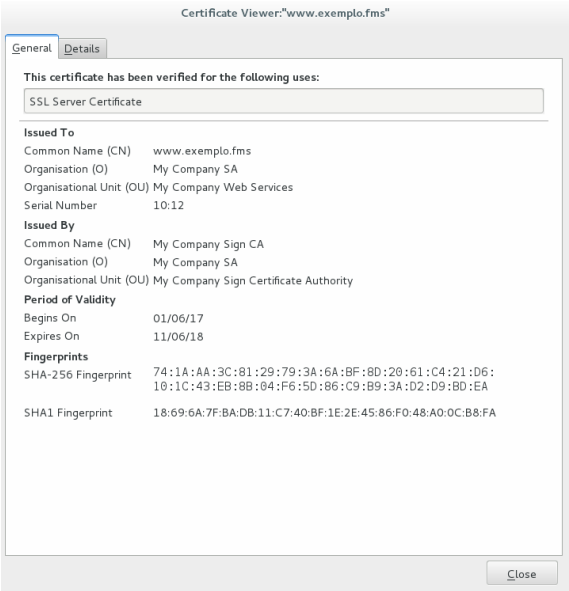


Figura B.3: Certificado da página *www.exemplo.fms*

A transformação dos acessos à página para HTTPS realizou-se através de um conjunto de alterações do ficheiro `/etc/apache2/sites-available/exemplo.conf`, tais como a indicação da porta 443, a ativação do modo SSL, a indicação da localização do certificado, entre outras.

A figura B.4 representa o acesso via navegador *web* à página criada, onde é possível confirmar o acesso HTTPS.

Figura B.4: Acesso à página *www.exemplo.fms*

B.4 Implementação Registo TLSA

A inclusão de um registo TLSA num servidor DNS requer, numa primeira instância, a geração do respetivo registo. Para isso, foi utilizada a página *web*, *Generate TLSA Record*[1], que possibilita a criação de registos TLSA mediante a apresentação do certificado e dos parâmetros que permitem identificar o tipo de utilização do registo.

Desta forma, para o registo foram definidos o campo de utilização como 1 (PKIX-EE), o campo de seletor como 0 (Cert), o campo de tipo de correspondência como 1 (SHA-256 *hash*). Todos estes campos encontram-se explicitados no capítulo 6 deste documento.

O resultado do processo anterior consiste no registo TLSA correspondente aos parâmetros anteriores:

```
_443._tcp.www.exemplo.fms.  IN TLSA 1 0 1
741aaa3c8129793a6abf8d2061c421d6101c43eb8b04f65d86c9b93ad2d9bdea
```

O passo seguinte à criação do registo consistiu na sua implementação no servidor DNS autoritário localizado na máquina *Debian 5*. Para isso, foi inserido o registo TLSA no ficheiro `db.exemplo.fms` localizado no diretório `/etc/bind` seguido da instrução referente ao comando *zonesigner*, de modo a atualizar os registos DNSSEC da zona.

A execução do seguinte comando na máquina *Debian 4* serviu como verificação para a correta instalação do registo:

```
# dane -rfc www.exemplo.fms
```

Resultado do comando anterior:

```
_443._tcp.www.exemplo.fms IN TLSA 1 0 1
741AAA3C8129793A6ABF8D2061C421D6101C43EB8B04F65D86C9B93AD2D9BDEA
```

O seu resultado confirmou que o registo foi implementado com sucesso.

[1] https://www.huque.com/bin/gen_tlsa

B.5 Análise Wireshark

Para a análise, procedeu-se à ativação da ferramenta *Wireshark* de modo a escutar as interfaces de rede da máquina *Debian 4*. Nessa mesma máquina foi realizado um acesso ao domínio *www.exemplo.fms* através do navegador *web Firefox*.

A primeira verificação realizada consistiu na consulta do protocolo DNS para investigar se o registo TLSA do domínio em questão foi obtido pela máquina. Na figura B.5 é possível verificar a existência da consulta por parte da máquina ao servidor DNS do registo TLSA correspondente ao domínio *www.exemplo.fms* e a existência de uma resposta indicando os vários parâmetros do registo TLSA.

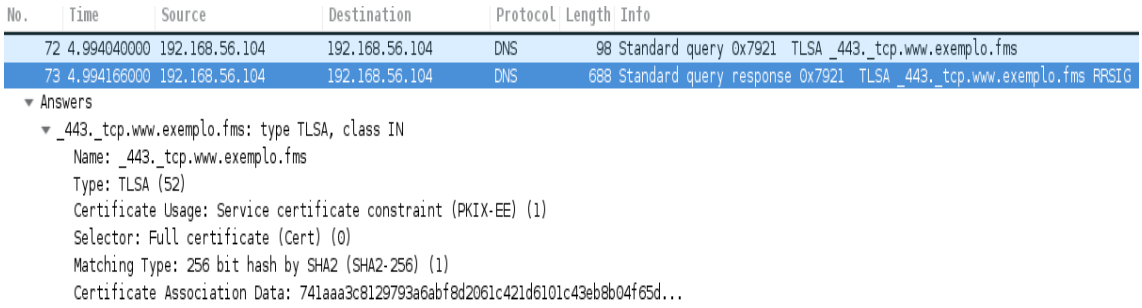


Figura B.5: Comunicação DNS referente ao registo TLSA

A segunda verificação consistiu na análise do protocolo TLS de modo a validar o certificado recebido pela máquina no acesso ao domínio. Na figura B.6 é possível verificar a mensagem transmitida pelo servidor localizado na máquina *Debian 6* (192.168.56.106) à máquina *Debian 4* (192.168.56.104) com os detalhes do certificado em questão inseridos na comunicação do protocolo de *handshake*.

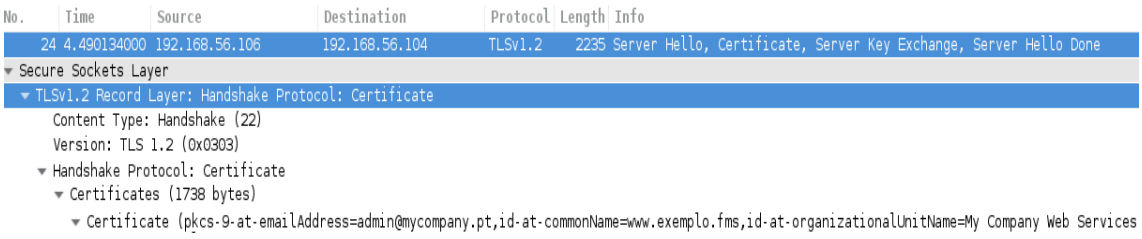


Figura B.6: Comunicação TLS referente ao certificado do domínio

Transformação do tipo de certificado

Através da ferramenta *Wireshark* foi possível obter o certificado no formato `der`, o que obrigou a realizar uma alteração de formato para `pem` através do seguinte comando:

```
# openssl x509 -inform der -in exemplo.fms.der -out exemplo.fms.pem
```

Desta forma foi gerado o ficheiro *exemplo.fms.pem* através do ficheiro **der** descarregado através do *software Wireshark*.

Referências

[1] Ivan Ristic. *Bulletproof SSL and TLS*. Feisty Duck, Londres, Agosto 2014.

[2] Tutorials Point. Cryptography hash functions. URL: https://www.tutorialspoint.com/cryptography/cryptography_hash_functions.htm. Último acesso em 27-12-2016.

[3] Suranjan Choudhury. *Public key infrastructure : Implementation and Design*. M&T Books, Nova York, 2002.

[4] W3Techs. Market share trends for ssl certificate authorities for websites. URL: https://w3techs.com/technologies/history_overview/ssl_certificate, 2016. Último acesso em 05-01-2017.

[5] William Stallings. *Cryptography and Network Security : Principles and Practice*. Prentice Hall, Boston, Mass. ; Londres, 5 edição, 2011.

[6] D. Cooper, NIST, S. Santesson, Microsoft, S. Farrell, Trinity College Dublin, S. Boeyen, Entrust, R. Housley, Vigil Security, W. Polk, e NIST. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. RFC 5280, maio 2008. URL: https://tools.ietf.org/html/rfc5280#section-4.2.

[7] Y. Pettersen. The Transport Layer Security (TLS) Multiple Certificate Status Request Extension. RFC 6961, junho 2013. URL: https://tools.ietf.org/html/rfc6961.

[8] P. Hoffman, VPN Consortium, J. Schlyter, e Kirei AB. The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA. RFC 6698, agosto 2012. URL: https://tools.ietf.org/html/rfc6698.

[9] P. Mockapetris e ISI. Domain Names - Concepts and Facilities. RFC 1034, novembro 1987. URL: https://www.ietf.org/rfc/rfc1034.

[10] ICANN. Dnssec – what is it and why is it important? URL: https://www.icann.org/resources/pages/dnssec-qaa-2014-01-29-en. Último acesso em 05-04-2017.

[11] ICANN. About icann. URL: https://www.icann.org/resources/pages/welcome-2012-02-25-en. Último acesso em 07-02-2017.

[12] ICANN e VeriSign. Root dnssec. URL: http://www.root-dnssec.org/. Último acesso em 19-04-2017.

[13] CMU CyLab. Perspectives project. URL: https://www.cylab.cmu.edu/partners/success-stories/perspectives.html. Último acesso em 16-01-2017.

[14] Perspectives-project. What is perspectives? URL: https://perspectives-project.org/. Último acesso em 16-01-2017.

[15] Moxie Marlinspike. Ssl and the future of authenticity. Em *Black Hat USA*, 2011. Disponível em https://www.rsaconference.com/writable/presentations/file-upload/ht2-107.final.pdf.

[16] Moxie Marlinspike. Software. URL: https://moxie.org/software.html. Último acesso em 29-01-2017.

[17] Peter Eckersley. Sovereign keys: A proposal to make https and email more secure. URL: https://www.eff.org/deeplinks/2011/11/sovereign-keys-proposal-make-https-and-email-more-secure. Último acesso em 30-03-2017.

[18] Certificate Transparency. Certificate transparency. URL: https://www.certificate-transparency.org/what-is-ct. Último acesso em 02-02-2017.

[19] C. Evans, C. Palmer, R. Sleevi, e Inc. Google. Public Key Pinning Extension for HTTP. RFC 7469, abril 2015. URL: https://tools.ietf.org/html/rfc7469.

[20] Mozilla Developer Network. Http public key pinning (hpkp). URL: https://developer.mozilla.org/en-US/docs/Web/HTTP/Public_Key_Pinning. Último acesso em 13-02-2017.

[21] Moxie Marlinspike. Trust Assertions for Certificate Keys. Internet-Draft draft-perrin-tls-tack-02, Internet Engineering Task Force, Janeiro 2013. Work in Progress. URL: https://datatracker.ietf.org/doc/html/draft-perrin-tls-tack-02.

[22] J. Callas, PGP Corporation, L. Donnerhacke, IKS GmbH, H. Finney, D. Shaw, e R. Thayer. OpenPGP Message Format. RFC 4880, novembro 2007. URL: https://tools.ietf.org/html/rfc4880.

[23] C. Ellison, Intel, B. Frantz, Electric Communities, B. Lampson, Microsoft, R. Rivest, MIT Laboratory for Computer Science, B. Thomas, Southwestern Bell, T. Ylonen, e SSH. SPKI Certificate Theory. RFC 2693, setembro 1999. URL: https://www.ietf.org/rfc/rfc2693.

[24] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Consulted*, 1:2012, 2008.

[25] Andreas Loibl. Namecoin. Em *Proceedings of the Seminars Future Internet (FI) and Innovative Internet Technologies and Mobile Communications (IITM), Summer Semester 2014*. Departmento de Ciências da Computação, Universidade Tecnica de Munique, agosto 2014. URL: https://www.net.in.tum.de/fileadmin/TUM/NET/NET-2014-08-1/NET-2014-08-1_14.pdf.

[26] Muneeb Ali, Jude Nelson, Ryan Shea, e Michael J. Freedman. Blockstack: A global naming and storage system secured by blockchain. Em *Proceedings of the 2016 USENIX Annual Technical Conference*, Denver, CO, USA, June 2016. URL: https://www.usenix.org/node/196209.

[27] Carl Ellison e Bruce Schneier. Ten risks of pki: What you're not being told about public key infrastructure. *Computer Security Journal*, 16(1):4–7, 2000.

[28] Douglas Perry. Diginotar breach affected 531 certificates, Setembro 2011. URL: http://www.tomsguide.com/us/diginotar-ca-data-breach-comodohacker,news-12421.html. Último acesso em 25-01-2017.

[29] Mike Wood. Fraudulent certificates issued by comodo, is it time to rethink who we trust?, Março 2011. URL: https://nakedsecurity.sophos.com/2011/03/24/fraudulent-certificates-issued-by-comodo-is-it-time-to-rethink-who-we-trust. Último acesso em 25-01-2017.

[30] UPORTO. A universidade do porto. URL: https://sigarra.up.pt/up/pt/web_base.gera_pagina?p_pagina=universidade. Último acesso em 28-04-2017.

[31] UPdigital. Segurança informática: Certificados digitais. URL: https://sigarra.up.pt/reitoria/pt/web_base.gera_pagina?p_pagina=1019447. Último acesso em 28-04-2017.

[32] OPENCA. Openca guide. URL: https://www.openca.org/projects/openca/docs/openca-guide.pdf. Último acesso em 15-05-2017.

[33] EJBCA. Ejbca - open source pki certificate authority. URL: https://www.ejbca.org/installations.html. Último acesso em 13-05-2017.

[34] Microsoft. Active directory certificate services step-by-step guide. URL: https://technet.microsoft.com/en-us/library/cc772393(v=ws.10).aspx. Último acesso em 19-05-2017.

[35] T. Ylonen, SSH Communications Security Corp, Ed. C. Lonvick, e Inc. Cisco Systems. The Secure Shell (SSH) Protocol Architecture. RFC 4251, janeiro 2006. URL: https://www.ietf.org/rfc/rfc4251.

[36] V. Dukhovni, Two Sigma, W. Hardaker, e Parsons. SMTP Security via Opportunistic DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS). RFC 7672, outubro 2015. URL: https://tools.ietf.org/html/rfc7672.

[37] BBN Technologies Richard L. Barnes. Domain name authentication with dnssec and dane. *The Internet Protocol Journal*, 15(1):12–23, março 2012.

[38] IANA. Root servers. URL: https://www.iana.org/domains/root/servers. Último acesso em 10-04-2017.

[39] APNIC. Dnssec validation rate by country. URL: https://stats.labs.apnic.net/DNSSEC. Último acesso em 12-04-2017.

[40] APNIC. Use of dnssec validation for world. URL: https://stats.labs.apnic.net/dnssec/XA. Último acesso em 12-04-2017.

[41] Internet Society. State of dnssec deployment 2016. Relatório técnico, Internet Society, dezembro 2016. Disponível em https://www.internetsociety.org/sites/default/files/ISOC-State-of-DNSSEC-Deployment-2016-v1.pdf.

[42] Internet Society. cctld dnssec status. URL: http://www.internetsociety.org/deploy360/wp-content/uploads/2013/04/2016-12-12-2016-12-12.png. Último acesso em 15-04-2017.

[43] Verisign Labs. Deployment growth. URL: http://secspider.verisignlabs.com/growth.html. Último acesso em 16-05-2017.

[44] Adam Langley. Why not convergence? URL: https://www.imperialviolet.org/2011/09/07/convergence.html, setembro 2011. Último acesso em 24-05-2017.

[45] Certificate Transparency. How certificate transparency works. URL: https://www.certificate-transparency.org/how-ct-works. Último acesso em 01-05-2017.

[46] Certificate Transparency. Certificate transparency - how log proofs work. URL: https://www.certificate-transparency.org/log-proofs-work. Último acesso em 05-04-2017.

[47] Bruce Morton. Certificate transparency deployment in 2017. URL: https://www.entrust.com/certificate-transparency-deployment-2017. Último acesso em 04-03-2017.

[48] Certificate Transparency. Certificate transparency - known logs. URL: https://www.certificate-transparency.org/known-logs. Último acesso em 05-04-2017.

[49] Qualys. Ssl report: sigarra.up.pt. URL: https://www.ssllabs.com/ssltest/analyze.html?d=sigarra.up.pt. Último acesso em 22-04-2017.

[50] P. Hoffmani, ICANN, J. Schlyter, e Kirei AB. Using Secure DNS to Associate Certificates with Domain Names for S/MIME. RFC 8162, maio 2017. URL: https://tools.ietf.org/html/rfc8162.

[51] P. Wouters e Red Hat. DNS-Based Authentication of Named Entities (DANE) Bindings for OpenPGP. RFC 7929, agosto 2016. URL: https://tools.ietf.org/html/rfc7929.

[52] Peter Gutmann. Engineering security. Book Draft, abril 2014.